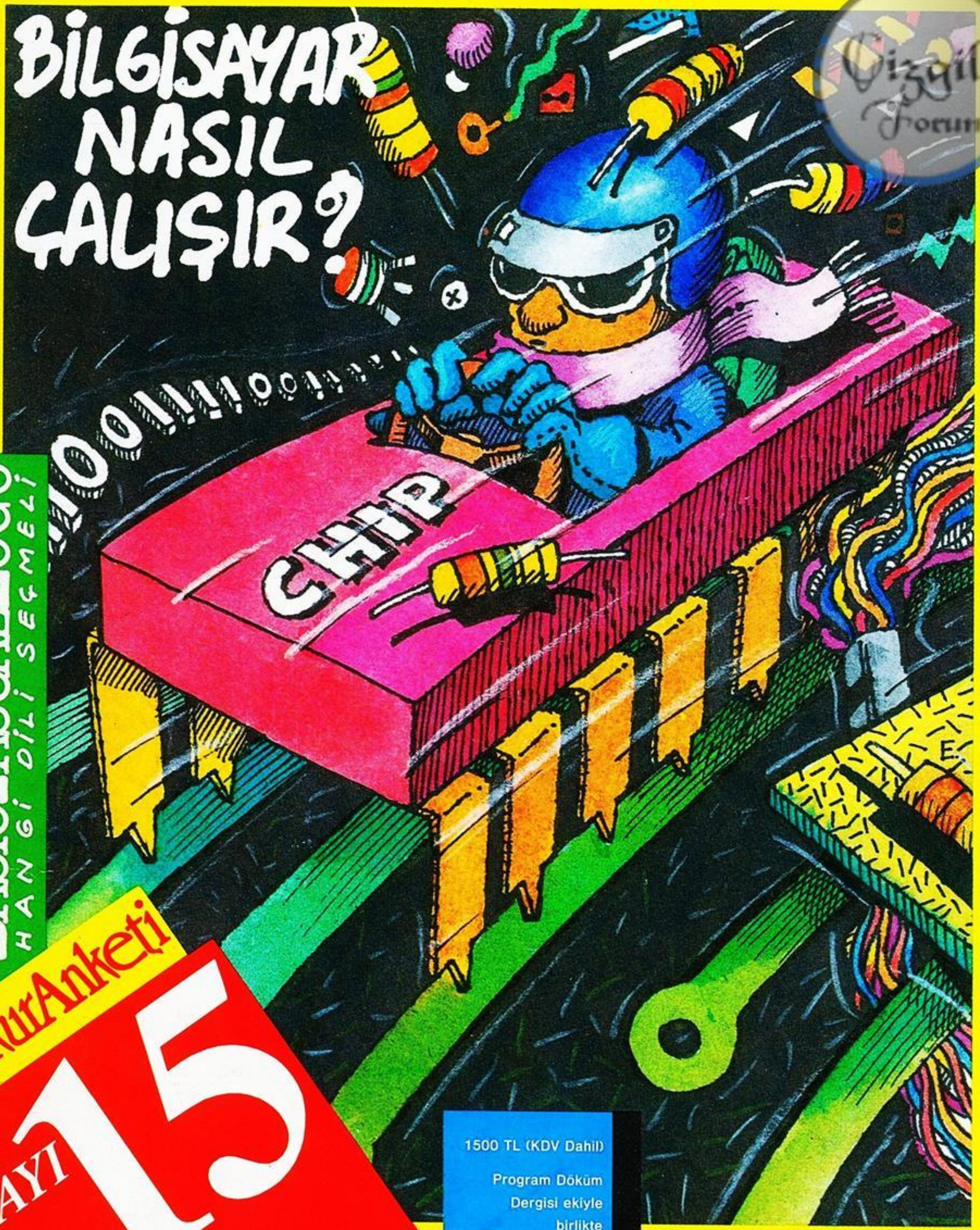
COMMODORE

AYLIK BİLGİSAYAR DERGİSİ Mayıs 1987

Özgürlü Forum

BİLGİSAYAR NASIL ÇALIŞIR?



yolbulduran-cdemir

BASIC PASCAL LOGO
HANGİ DİLİ SEÇMELİ?

OkurAnketi

SAYI 15

1500 TL (KDV Dahil)

Program Döküm Dergisi ekiyle birlikte

Dünyanın en kolay kelime işlem paketi "Easy" şimdi Türkçe

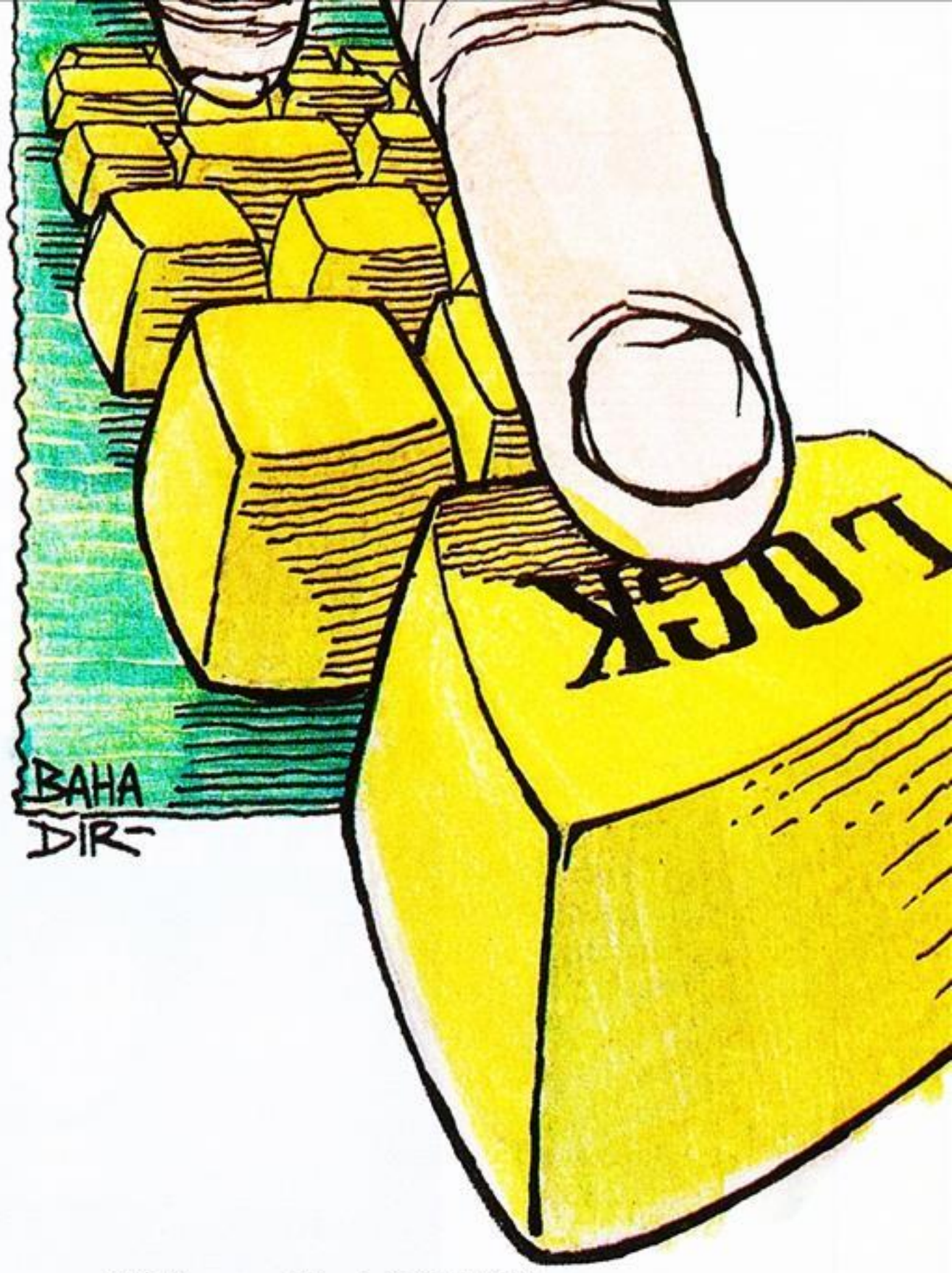
Artık bilgisayarınıza* Türkçe komut verebilirsiniz. Komdata, MikroPro'nun kullanımı en kolay ve en yetenekli kelime işlem paketini Türkçeleştirdi. **Bu paketi kullanmak neden bu kadar kolay?** 1. Hiçbir komutu ezberlemenize gerek bırakmayan menüler içerir. 2. Herhangi bir komutu kullanırken, o komut hakkında ayrıntılı bilgiyi anında görüntüler. 3. Her işlem sırasında sizi adım adım yönlendirir. 4. Yanlışlıkla silinen bir satırı veya paragrafı geri getirebilir. 5. Sınırsız uzunlukta ve 240 sütun genişliğinde dokümanlar yaratmanıza izin verir. 6. Wordstar ve Wordstar 2000 ile yazılan dokümanları kullanabilir. 7. Lotus 123 ve Symphony dosyalarını doğrudan okur. 8. 250'den fazla yazıcının özelliklerini tanır. **Klavyenizle Türkçe komut vermenin rahatlığına biran önce kavuşmak için bizi 1330342'den hemen arayın.**

*Commodore PC ve uyumluları

KOMData

KOMDATA İLETİŞİM SİSTEMLERİ LTD. ŞTİ.

İçindekiler



DİLLER

Basic mi? Pascal mı? Logo mu? 6
Sizin için en uygun dil hangisi?

Logo 12
Artık sadece bir çocuk dili olmaktan çıkan Logo, içerdiği recursion yeteneği sayesinde çok çeşitli problemlerin çözümü için ideal bir dil haline gelmiş durumda.

Makine Diline Merhaba (5) 26
Bu bölümle birlikte artık makine dilinin bütün komutlarını öğrenmiş olacaksınız. Komutların tam listesi için Döküm Eki'ne bakınız.

Pascal (3) 64
Yapılandırılmış programlamanın bu klâsik dili hakkındaki yazı dizimiz sürüyor.

YAZILIM

C-128 için Püf Noktaları (2) 44
Bilgisayarınızdan daha verimli bir biçimde yararlanmanızı sağlayacak yeni ve kolay püf noktaları.

C-64 ile Grafik (7) 48
Dizinin bu bölümünde kendi istediğiniz karakterleri tanımlamanızı mümkün kılacak bir karakter editörü programının nasıl yapılacağı anlatılıyor.

TOPLUM

Bilgisayar ve Toplum 15
Ram Dergisi'nin Cumhuriyet Kitap Kulübü'nün Kitap Şenliği çerçevesinde düzenlediği bu panelde, bilgisayar çağının Türkiye için yarattığı sorunlar ve önüne açtığı ufuklar tartışıldı.

Bilgisayarlı Eğitimden Ne Haber? 30
Bilgisayar devriminin beşiği Amerika'da, bilgisayar eğitimi ne yönde ve nasıl dönüştürüyor?

Bilgisayar Destekli Eğitime Girerken 62
Osman Ongun, önümüzdeki yıllarda kamuoyunun gündemini işgal etmesi gereken en önemli konulardan biriyle ilgili olarak nelerin gözönünde bulundurulması gerektiğini tartışıyor.

TANITIM

Bu ayki tanıtım sayfalarınızda, iki oyunun (**Gunship** ve **Tao Ceti**) yanısıra, C-64 için geliştirilen yeni işletim sistemi **Geos** tanıtılıyor. 20

DONANIM

Bilgisayar Nasıl Çalışır? 34
Commodore 64'ün bütün temel bileşenleri ve işlevleri.

Ses ve Müzik için Digitizer 53
Bilindiği gibi ses dalgaları analog bilgi taşıyan uyarıcılardır. Bunların bilgisayar tarafından işlenebilmesi için önce sayısallaştırılmaları gerekir. Bu işlemi gerçekleştirecek bir gerecin nasıl yapılabileceği anlatılıyor.

UEM

SÜPER IŞIKLI KALEM

Atari 800 XL

Amstrad CPC 464 Renkli Monitörlü

Sinclair Spectrum 48K, 48K+

Commodore 64

Fiyatı: 10.000 TL

UEM

SÜPER IŞIKLI KALEM

Tüm Commodore Satıcılarında

80220 Halaskârgazi Cad. Pekin Apt. 210/3 Osmanbey/İST. Tel: 147 03 54

COMMODORE
AYLIK BİLGİSAYAR DERGİSİ Mayıs 1987

Teleteknik A.Ş. Adına
Sahibi ve Sorumlu Yazışleri Müdürü
OSMAN BABAOĞLU

Yayın Yönetmeni
İSKENDER SAVAŞIR

Yayın Koordinatörü
TURHAN TANDOĞMUŞ

Danışma Kurulu
FATİH ARIMAN
BİLAL SADİ OĞUZ
EMRE SENAN
ÜMİT KIVANÇ

Yayın Kurulu
KENAN AKKERMAN
TUNA ERTEMALP
MUSTAFA GÜVEN
OSMAN HIZAL
AYHAN KALAYLIOĞLU
SAMET KUMAŞ
TAN ORAL
AFİF SAY
AYŞEGÜL TARIM

Teknik Üretim
PERKA A.Ş.

Dizgi
MARATON DİZGİEVİ

Fotoğraflar
MEHMET ÖZCAN

Reklam ve Halkla İlişkiler
AYSIN AYDEMİR

Reklam Rezervasyonları
145 13 30 (4 HAT)

Katkıda Bulunanlar
BİRGÜL ASENA
BİLGE AYAN
SİLVA AVEDİKOĞLU
SEMRA ÇAĞLAYAN
SEYFETTİN ÇANKAYA
ÖJENİ GÜLLAPYAN
FETHİ SERİM
ANKİN SİNANYAN
AZİZ ŞENLEN
MAHMURE ÜNYELİ

Renk Ayrımı
EBRU GRAFİK

Baskı
ADSAN A.Ş./MERTER

Dağıtım
GAMEDA

Teleteknik A.Ş.
Merkez
RIHTIM CAD. NESLİ HAN NO: 207/4,
80030 KARAKÖY-İSTANBUL
TEL: 145 13 30 (4 HAT)
TELEKS: 25731 TLTK TR
Bilgisayar Danışma Merkezi
SİLAHHANE CAD. RALLİ APT.
59/3 80200 TEŞVİKİYE-İSTANBUL
TEL: 131 15 49-50

YAYINCININ NOTU

MÜJDELERLE DOLU BİR SAYI

Bu sayımızın en önemli özelliği, bize birbiri üstüne müjdeler iletme olanağını tanıması. Bunlardan birincisi, derginizin sayfalarını karıştırırken kendiniz göreceksiniz. Artık Commodore'un altmışdört sayfası da renkli olarak çıkıyor.

Ama daha da kıvandırıcı olan ikinci müjdemiz: İncelemeleri sonucunda Milli Eğitim Bakanlığı Commodore dergisinin ilk beş sayısının Tebliğler Dergisi'nde duyurulmasını uygun görmüş. Yola çıkarken asık yüzlü olmadan eğitici olabilecek bir yayın yapabilmeyi umuyorduk. Bu karar, kısmen de olsa, başarılı olduğumuzu gösteriyor.

Size ileteceğimiz son haberin biraz buruk bir yanı da var. Geçtiğimiz sayı, teknik hazırlığını Boyut A.Ş.'nin yaptığı son sayıydı. Commodore Dergisi'nin doğumunda hazır bulunan Boyut A.Ş. çalışanlarının, Saadettin Oktay'ın, Bülent Özükan'ın dergiye olan katkılarını ifade edebilecek kelimeleri aramak beyhude bir çaba olurdu. Tabiî, teknik yönetmenimiz Sevinç Yenal'ın yeri apayrı... Bizim gözümüzde dergiyle o kadar içiçe geçmişti ki, bugün eski sayıların sayfalarını çevirirken neredeyse Sevinç'in yüzünü görür gibi oluyoruz. Sevinç'e ve bütün diğer Boyut çalışanlarına, gösterdikleri sabır için, birlikte çalışmayı bir zevk haline getirdikleri için teşekkür eder, başarılar dileriz.

Boyut'tan ayrılmanın burukluğuyla elinizde tutmakta olduğunuz sayıyı hazırlamanın pek o kadar zevkli olmayacağını düşünüyorduk. Tâ ki, Boyut'un yerini alan Perka A.Ş.'nin çalışanlarıyla karşılaşıncaya kadar. Derginin yeni teknik yönetmeni Ümit Kıvanç Milliyet'ten, Cumhuriyet'ten YeniGündem'e kadar basın yayın dünyasını çok farklı konumlardan tanımış biri. Şefika Fırat'la birlikte çalışmalarımıza yeni bir soluk, zaman zaman bizi ürkütse de heyecanlandıran bir disiplin anlayışı getirdi.

Umarım bu değişikliğin sayfalarımıza yansıyan sonucundan siz de memnun kalacaksınız. □

OSMAN BABAOĞLU

Hangi Dili Seçmeli?

BASIC Pascal LOGO

Bir sonraki bilgisayar kuşağında BASIC yerine Pascal'ın ön plana geçeceğini düşünen uzmanların sayısı gitgide artıyor. O halde, bu diller hakkında daha fazla bilgi sahibi olmakta yarar var.

Belçika'da göstericilerin sokağa dökülmeleri bir hükümet bunalımına yol açtı; hattâ Hindistan'da ölümle sonuçlanan olaylar çıktı; Alman yüksek öğretim kurumlarında ise akademik bir bayrak altında amansız ve çılgınca bir mücadele başlatıldı. Her üç olayın temelinde de aynı neden yatmaktaydı: Dil ile ilgili bir kavgaydı bu. Ancak, Belçika'da ve Hindistan'da patlak veren olaylarda halkın hayatî önem taşıyan çıkarları gündeme gelirken, Almanya'daki gösterilerin odak noktasında, bilgisayar dersinde hangi programlama dilinin daha uygun olduğu sorunu yer almaktaydı.

Bu kavgayı bugüne değin hiçbir taraf kazanamadı. Doğrusu da budur; çünkü en iyi dil diye bir şey yoktur. Dillerden bazılarının belirli bazı amaçlar için daha uygun oldukları söylenebilir olsa olsa.

Biz bu konuda asıl neyin önemli olduğunu göstermeye çalışırken, uzun kuramsal tartışmalara girmek yerine, somut örneklerden yola çıkmayı tercih ettik. Bu nedenle de, sizlere sunmak üzere ele aldığımız bazı karakteristik programlama problemlerini, bilinen üç programlama dilinde çözmeyi uygun gördük.

Niçin BASIC, Pascal ve LOGO? diye soracaksınız. Bunun nedenlerini şöyle sıralamak mümkün: Çünkü mikro-bilgisayar alanında en yaygın olan ve hemen her ev-bilgisayarı sahibinin ilk el attığı dil BASIC'dir; çünkü Pascal birçoklarınca, algoritma işlemlerinin öğrenilmesinde uygun bir dil olarak kabul edilmekte ve profi-programlar alanında zevkle kullanılmaktadır; çünkü LOGO "çocuk dili"nin beraberinde getirdiği zevksizliği üzerinden atmıştır artık...

Son derece basit bir örnekle konuya girelim. Diyelim ki, bir veri-yönetim sistemi kurmak istiyoruz ve bunun için bir veri dosyasına gerek duymaktayız. Bu veri dosyası bir firmanın çalışanları ile ilgili olsun ve şu

Bir taslak, bir akış diyagramı

Boşver! Öylece yazıyorum işte!

Dur bakalım! Programlama dilin uygun mu acaba?

Şimdi biraz etraflıca düşünelim. Ne istiyoruz?

Dosya yönetimi ve biraz grafik...

İlkin bir ağaç resmi çizelim.

Komutları biliyor musun? PENDOWN, HIDETURTLE, FORWARD 100...

bölümleri içersin: Soyadı, Adı, Doğum tarihi, Posta kodu, Yer, Sokak, ilgilinin devamlı mı, geçici mi çalıştığı ve devamlı statüde ise, ücreti.

Böylece bilgisayarın başına geçtiğimi ve programı yazdığımı düşünün. Programı yazma, yanlış aramalarla ve düzeltimlerle birlikte, yaklaşık iki saatimi alıyor. Daha önceden hiçbir taslak yapmıyorum ama böyle bir lüksü ancak küçük programlarda ve zamandan tasarruf etmek amacıyla kendime tanıyabilirim. İşte şimdiden BASIC'in temel özelliklerinden biriyle tanışmış oluyoruz: Bu sevimli çevirici-dilde olduğu gibi, bu kadar kısa bir süre içinde hatasız ve kendi gereksinmelerine uyarlanmış küçük bir program herhalde başka hiçbir programlama diliyle yazılamaz. Öte yandan, programın kuruluşunun bir bakışta kavranamamasından doğan tehlikenin bu denli büyük olduğu bir başka programlama dili de yoktur. Bu noktada ona en çok yaklaşan diller Assembler ve Forth ve C gibi, Assembler benzeri dillerdir.

Hayatta hiçbir şeyin ideali olmadığı gibi, bilgisayar dillerinin de ideali yok. Seçilecek dil, yapılacak işe, beklentilere, hattâ zevke bağlı.

Pascal hayranı olan bir mesai arkadaşım bu programı önce Pascal diliyle yazdı. Her iki mini-programı karşılaştırdığınızda, iki şey hemen gözünüze çarpacaktır kuşkusuz: Pascal versiyonu 1) çok daha uzundur, 2) bir bakışta çok daha kolay kavranabilir.

Pascal'ı güçlü kılan özelliklerden biri -ki bunu herhalde bilirsiniz- onun "yapısallaştırma" yetisidir. Bu program-yapısı daha dökümde, yani programın simgesel ifadesinde apaçık görülür. Bu sayede, daha sonra yapılan düzeltimlere ve katkılara kolaylık sağlanmış olmaktadır. Böylece, karmaşık tarzda programlar da okunabilirliklerini korumuş olurlar. Ayrıca, Pascal'ın pedagojik etkisini de küçümsememek gerekir: Pascal, dakikliğiyle ve her değişkenin kullanımdan önce tam olarak açıklanmasını şart koşmasıyla, programcıyı disipline ve özene zorlar.

Peki "yapısallaştırma" uygulamada ne anlama gelmektedir? Bir gün şöyle bir deney yapmak zahmetine katlanmıştım: İçlerinde belli bir değerin hesaplandığı program satırlarını, yabancısı olduğum bir BASIC programında aramaya çalışıyordum. Bir süre geçtikten sonra üzülerek gördüm ki, satırlar programın tümüne dağılmışlardı. Buna GOTO ve GOSUB komutları yol açmıştı. Gerçi GOTO komutu Pascal'da da var, ama orada pek makbul sayılmıyor ve birçok durumda da zaten gereksiz. Bunlar orada bloklar halinde çalışırlar ve bloklar içindeyse gerçekten birbirlerine ait olanlar bir araya getirilmişlerdir. Bu tür bloklar ya yordamlar (procedure) ve fonksiyonlardır (bunları "adları olan alt-programlar" olarak tanımlamak mümkündür), ya da bir **"begin"** komutuyla başlayan ve bir **"end"** komutuyla sona eren başka program birimleridir.

Profesyonel bir Pascal-programının oldukça tuhaf bir görünümü vardır. Programın ilk % 90'ı sadece tanımlar içerir; değişkene, yordamlara ve fonksiyonlara ilişkin tanımlar... Asıl program ise, içlerinde bu yordamların harekete geçirildiği birkaç satırdan oluşmaktadır. Böylelikle, programın okunmasına yüksek bir düzeyden -ana program düzeyinden- başlanabilmektedir. Daha sonra da ve ancak tedrici olarak programın diğer bölümlerine nüfuz edilebilmektedir. Tabii bunun tersi bir yöntem de geçerlidir: Programlamada ilkin programın kaba bir taslağıyla işe başlanmakta ve programa yavaş yavaş incelik kazandırılmaktadır. Buna "Top-Down" yöntemi denilmektedir. Adından da anlaşılacağı üzere, bu yukardan aşağıya doğru uygulanan bir yöntemdir.

Ama "yapısallaştırma" sadece program yapısıyla sınırlı olmayıp bütün veri tiplerine yayılabilir. Böylece Pascal, BASIC'de bulunan verilerin ötesine geçmenin sözkonusu olduğu her yerde uygun bir programlama dili olarak sunar kendini bize. Örneğin doğum tarihi için bunu söyleyebiliriz: BASIC'de doğum tarihini ya üç sayıya ayırmak, ya da bu tarihi sözkonusu üç sayının kendisinden güçlükle çıkarıldığı bir karakter zinciri olarak işlemek zorunluğu vardır. Pascal'da ise üç rakamdan oluşan "Tarih" adlı yeni bir veri tipini tanımlamaktan başka bir şey yapmaya gerek yoktur. Bu arada, anılan üç rakamdan her birinin belli sınırlamalara tâbii olduğunu belirtmek gerekir. Böylece, doğum tarihi daha anlaşılır olmakta ve daha kolay okunabilmektedir. Bu sayede ayrıca, programdaki bazı işlemler daha kısalmakta ve hızlanmaktadır.

Ele aldığımız örnekteki veri yığını da böylelikle temel veri tiplerinin bir araya getirilmesiyle oluşmuş bulunuyor: Tam sayılar (INTEGER), onda-

lık sayılar (REAL), karakterler (CHAR), karakter zincirleri (STRING) ve mantıksal değişkenler (BOOLEAN). Bunların dışında başka her şeyi kendi gereksinmelerinize göre ve kendi kendinize oluşturmanız mümkündür.

Ama ben BASIC-sever bir kişi olarak Pascal'ın yolaçtığı güçlükleri düşünmeden edemiyorum. Bu dil pek çok alanda fazla muğlak geliyor bana. Kendi BASIC programlarımı zengin ve anlaşılır açıklamalar sayesinde her zaman okuyabiliyorum. Nasıl bir disiplin uygulanması gerektiğini de, bana dikte ettirilmesindense, ken-

Bilgisayarla yapılmasını istediğimiz işin niteliği kadar, çalışma tarzımız da dil seçiminde etkili oluyor.

dim belirlemeyi tercih ediyorum. Ayrıca, Pascal'daki tipler zorunluluğunun sıkılığı kimi zaman gerçekten eziyetli oluyor. Sonuç olarak bir sayının INTEGER mi (tam sayı) yoksa REAL mi (ondalık sayı) olduğunu bazen bizzat bilgisayar belirlemelidir. Hem zaten o bunu benden daha iyi yapabilir. Tiplerin örneğimizdeki gibi güzelce bildirilmiş olmasına gelince, eğer bu bana bir avantaj sağlamıyorsa ne yararı vardır bu bildirimin? Çünkü belirlenen sınırlar dışında bilgisayara veri aktarılması gene de mümkündür ve Pascal hiçbir hatayı bildirmemektedir. Yalnızca Pascal-derleyicisi **(Compiler)** bu konuda bir avantaj sağlayabiliyor; çünkü bu sayede kendi depo alanı düzene sokulabilmektedir.

LOGO'NUN BÜYÜK GÜCÜ: RECURSİON

Bir keresinde bizim yazı kurulumuzda -başlangıçta çocuklar için geliştirilmiş- LOGO programlama diliyle ilgili olarak şiddetli bir tartışma çıkmıştı. Taraflardan biri LOGO'nun bütünüyle BASIC'le ya da Pascal'la kıyaslanabileceğini, dolayısıyla da LOGO ile yararlı programlar yapılabileceğini öne sürüyordu. Öbür grup ise LOGO'nun hâlâ "çocuksu" olduğunu ve bu yolla -güzel grafik çizimler dışında- aklı başında hiçbir şey yapılamayacağını iddia ediyordu. Bu tartışmada herhangi bir çözüme ulaşılamadı. Biz de örnek programımızı LOGO dilinde yazmaktan kaçındık. Giriş/Çıkış rutinlerini ve veri yönetimlerini programlamak çok zahmetli bir işti. Çünkü bu dilin gücü, ona kızkardeşi LISP'ten ("yapay zekâ" için öngörülen programlama dili) miras kalan bir özellik olan "recursion"dan kaynaklanmaktaydı. (Recursion Pascal'da da vardır.)

Bir ağaç resmi çizmek istediğinizi düşünün. Önce 10 cm. uzunluğunda bir cetvel alın ve bir çizgi çekin. Daha sonra cetveli, diyelim yarı yarıya, kısa tutun, biraz döndürün ve işleme baştan başlayın. Bu işlem, bizim herhangi bir yerde bir sona ulaşıncaya değin gitgide küçülen değerlerle sürekli olarak tekrarlanmalıdır. (Çizimin nerede son bulacağını tabii sizin yerleştirmeniz gerekiyor programa.)

Gerçi şimdiden recursion'un özüne değin (aynı olayın sürekli tekrarlanması) bir fikir sahibi olduğunuz söylenebilir, ama tek başına bu kadar bilgi elbette yeterli değildir. BASIC'deki her FOR-NEXT de bunu pekâlâ yapabilir. Recursion'un önemli bir özelliği şudur ki, bilgisayara iletilen bir komut hemen uygulanamamakta, aynı şemaya göre bir dizi alt-komutun çözülmüş olması gerekmektedir.

Bunu somutlaştırmak için şöyle bir durumu gözünüzün önüne getirin: Bir dostunuz size geliyor ve bir bilgisayar edinmek istediğini bildiriyor. Ama amatör telsizci olan dostunuz bu hobisinden dolayı mutlaka özel bir program arzu ettiğini de sözlerine ekliyor ve size bu konuda kendisine bir tavsiyede bulunup bulunamayacağınızı soruyor. Siz herhangi bir tavsiyede bulunamıyorsunuz. Bununla birlikte, sizin de bilgisayar ticareti yapan bir tanıdığınız vardır ve bu kişiye amatör telsizciler için özel programı olan bir bilgisayar bilip bilmediğini soruyorsunuz. Ama o da yardımcı olamıyor bu konuda. Ancak, onun da danışacağı yetkili birisi vardır... vb. En sonunda bu konuda bilgi sahibi olan birini bulmak mümkün oluyor. Bu kişi doğrudan kendisine

BASIC

```
READY.
100 REM VERI KAYITLARI
110 DIM A$(8),B$(8)
120 FOR I=1 TO 8:READ
    A$(I):NEXT
130 DATA ''AD'',
    ''SOYAD'', ''DTARIH'',
    ''SOKAK'', ''SEMT'',
    ''SEHIR'',
    ''SUREKLI/GECICI'',
    ''UCRET''
140 PRINT ''♥'': PRINT
    TAB(10); ''VERI
    KAYITLARI'':PRINT
    TAB(10);
    ''= = = = = = = = = =
    = = = = = = ''
150 FOR I=1 TO 8:PRINT
    I;A$(I);'':'':INPUT B$(I):
    NEXT
READY.
```

Kısa ama karışık: BASIC'de veri soruşturması.

LOGO

```
TO S :X :ORD
  REPEAD 4 [CIZ :X :ORD
      SAGA :X] END
END

TO CIZ :X :ORD
  IF :ORD = 1 THEN
      FORM :X STOP
  CIZ :X :ORD — 1 SAGA :X
  CIZ :X :ORD — 1 SOLA :X
  CIZ :X :ORD — 1 SAGA :X
  CIZ :X : ORD — 1
END

TO FORM :X
  REPEAT 2 [FORWARD :X
      LEFT 45]
  FORWARD :X
END

TO SOLA :X
  LEFT 45
  FORWARD :X
  LEFT 45
END

TO SAGA :X
  RIGHT 90
  FORWARD :X
  RIGHT 90
END
```

''Sierpinski-eğrisi''ne ilişkin LOGO programı dört adet alt-rutine (SAGA, SOLA, FORM, CIZ) ve bir adet ana yordama sahip (S).

PASCAL

```
program dosya;
type tarih = record gün:          1..31;
                    ay:           1..12;
                    yıl:          1890..2000;
               end;
     adres = record ad:           string[80];
                    soyad:        string[80];
                    dogum:        tarih;
                    yer:          1000..8999;
                    semt:         string[80];
                    sokak:        string[80];
                    case denk:    boolean of
                          true:   (geçici: real);
                          false;  (  );
                    end;
var düzeltici: adres;
    sahte; string[80];
begin writeln ('                  bilgi girisi');
      writeln ('                  = = = = = = = ');
      with duzeltici do
      begin
      write ('ad.....:');readln (ad);
end;
end.
```

Buna tekabül eden Pascal programının uzun olduğu açık, ama ilk bakışta kavranması da çok daha kolay.

başvurmuş olana gerekli bilgiyi veriyor, bu bilgiyi alan da kendi tanıdığına bunu aktarıyor, o da öbürüne, tâ ki dostunuzun sormuş olduğu şey kendisine ulaşıncaya kadar.

Recursion mekanizması en basit durumda böyle işlemektedir. Ama örneğin bir kitabın çözümlenmesi sözkonusu olduğunda karmaşık durumlar ortaya çıkabiliyor. Kitap önce bölümlere, bölümler cümlelere, her bir cümle dilbilgisel kategorilere, her kategori alt-kategorilere, her alt-kategori kelimelere, her kelime hecelere ve her hece de harflere ayrıştırılıyor. Ayrıştırma süreçlerinin hepsi birbirine benzemektedir. Ama bir ayrıştırma işlemi her zaman ânında tamamlanmamakta, muhtemelen de bir sonraki evrenin çözümlenmesine bağımlı kalmaktadır.

LOGO'nun somut diline çevrildiğinde bunun anlamı şudur: En basit rekürsif fonksiyon -belki de okul dönemlerinden hatırlayacağınız- faktöriyel işlemdir. Bu işlem, sayının ardından gelen bir ünlem işaretiyle ifade edilmektedir: $3! = 1 \times 2 \times 3 = 6$. Ancak, rekursif olarak bu fonksiyon şöyle tanımlanmaktadır: N sayısının faktöriyeli, ilkin N-1'in bulunması ve bunun da N sayısı ile çarpılması yoluyla hesaplanmaktadır. Bunun içinse tabii önce (N-1)! sayısının bilinmesi gerekmektedir ki, bu da (N-2)den çıkarılmaktadır... vb. Ta ki N=1'e ulaşıncaya kadar. Çünkü matematikçiler 1'in faktöriyel değerinin 1 olduğunu saptamışlardır: 1! = 1.

LOGO'da ise bu şöyle ifade edilmektedir:

```
TO FAK = N
IF: N=1 THEN OUTPUT 1
STOP
ELSE OUTPUT: N*FAK (: N-1)
```

Bu formülü şöyle açıklayabiliriz: Burada ''N'' N'in değeridir. ''OUTPUT'' ise bir şeyin geri verileceğini göstermektedir. Çünkü her LOGO-yordamı tek bir görevi yerine getirir. Bu görevi dolaysız olarak yerine getiremediği takdirde, komutu bir alt-programa iletir. Bu alt-program sözkonusu komutun yordam tanımını temin eder ve sorunu çözmeye çalışır. Eğer başarılı olamazsa, o da tıpkı bir önceki gibi yaparak komutu bir alt-alt-programa aktarır. Bu işlem, komut vericilerinden biri istediği sonucu alıncaya kadar böylece sürüp gider. Bu durumda sonuç bir önceki komut vericiye iletilir (OUTPUT talimatı aracılığıyla). Bu da bir kez daha görevini yerine getirmeyi dener.

Bunda başarılı olduğu takdirde, elde ettiği sonucu kendi komut vericisine (yukarıya) aktarır... vb. En sonunda ise, eğer bütün komut vericiler kendi görevlerini yapmışlarsa, LOGO basıma geçebilmektedir.

Bu anlatılanlar karmaşık mı geliyor size? Öyle olsa da, kaygılanmanıza gerek yok. Çünkü "recursion" veri işlemin en tuhaf ve güç anlaşılır kavramlarından biridir. Bilgisayarın bütün hesap işlemlerine saydam bir şekilde ışık tutacak basit bir örnekle konuya açıklık getirmeye çalışalım: Örneğin, 3 sayısının faktöriyelini hesapladığımızı düşünelim. Bu durumda LOGO ilkin N = 1 olup olmadığını sorar. Yanıt olumsuzdur. O zaman yordam (buna "1" diyelim) "3" değerini kaydeder ve bir alt-yordamı 3-1 = 2'nin faktöriyelini bulmakla görevlendirir. "2" olarak adlandıracağınız yordam ise N = 1 olup olmadığını yoklar; yanıt olumsuzdur ve yordam "2" bu defa 2 değerini kaydederken bir alt-alt-yordamı 2-1 = 1'in faktöriyelini bulmakla görevlendirir. Alt-alt-yordam "3"de N'in değerinin 1 olup olmadığını sorar; yanıt olumludur. Bu durumda yordam "3" büyük bir gururla yordam "2"ye sonucu aktarır. Şimdi yordam "2"nin kullanabileceği bir değeri vardır ve bunu $1 \times 2 = 2$ şeklinde kullanır. Bu sonuç ilk komut vericiye iletilir ve böylece o da tıpkı bir önceki gibi işlem yapabilecek bir değer edinmiş olurken bu (2) değerini kendi değeri olan (3) ile çarparak 6 sayısını bulur. Buysa, görüldüğü gibi, 3 sayısının faktöriyelidir: 3!

Şimdi artık önemli olanın ne olduğunu herhalde daha iyi anlaşılmaktadır: Sorun bizzat çözülemediğinde, görevi bir başkasına aktarmak. Bunun temelinde yatan ilkeyi "gecikmeli uygulama" olarak nitelendirmek mümkündür. Burada bütün ara değerler ve ara formüller depolanmak zorundadır ve bundan ötürü de rekursif diller çok fazla bellek hacmi gerektirirler. LISP'in, ev ya da kişisel bilgisayarlarda bu denli ender görülmesinin nedenlerinden biri de budur. Ama depolama konusunda programcının kaygıya kapılmasına hiç gerek yoktur; çünkü bütün bunları LOGO-çeviricisi üstlenmiştir.

Recursion'un ardında yatan ilke grafik olarak pek güzel gösterilebilir -"çocuk" dili LOGO'nun eğitsel yönden taşıdığı büyük değerin kaynağını da burada aramak gerekir. Bir örnek daha verelim: Geçen yüzyılın sonlarına doğru bazı matematikçiler son derece garip görünümlü birtakım eğriler keşfettiler. Bu eğrilere akademik çevrelerce "patalojik" damgası vuruldu. Çünkü bunlar, eğrilere ilişkin peşin yargıları altüst etmişlerdi. Sözkonusu "hastalıklı" eğriler bulundukları yüzeyi en sonunda baştanbaşa kaplıyorlardı; çünkü çok sık bir dallanma sürecine giriyorlardı. Öyle ki, her alt dal, dallanma sırasında yeni bir alt dal ve bu da başka bir alt dal... vb. oluşturuyordu. Tıpkı recursion'da olduğu gibi.

Bu yeni oluşumlar hiç de salt birtakım oyunlar sonucunda ortaya çıkmış değillerdi; birçok alana yeni bilgiler kazandıran ve olağanüstü bilgisayar grafiklerine imkân veren bir bilimin gelişmesi bu eğriler sayesinde olmuştur. Biz bu eğrilerden biri üzerinde çalışmaya başladık ("Sierpinski-eğrisi", adını bulucusundan almıştı). Eğriyi LOGO dilinde göstermeye uğraşıyorduk. Ben bir gün sonrasında sinirlerim harap olmuş bir durumda bu işten vazgeçerken, meslektaşım Lehmann, çalışmasını yılmadan sürdürdü ve rekursif hilelere karşı yürüttüğü toplam 24 saatlik bir mücadele sonucunda basit görünümlü bir program yapmayı başardı. Gerçekten de, LOGO'nun temel sözlük dağarcığına egemen olan herkes bu programı (anlamasa bile) "okuyabilirdi". Ama aynı program BASIC'de öylesine kapsamlı ve bir bakışta anlaşılması öylesine zordu ki, burada bunun dökümünü yapmaktan kaçınmak zorunda kaldık.

Kendi düşüncesi hakkında da düşünebilmek, insan zekâsının bir özelliği. Yapay zekâ dilleri bu özelliği Recursion aracılığıyla taklit eder.

Ayrıca, Pascal ile LOGO arasında bir karşılaştırma yapmaktan da çekindik. Çünkü LOGO'nun tipik grafik yetenekleri Pascal'da ya hiç yoktu, ya da aynı şekilde mevcuttu. Bu durum versiyona bağlı olarak değişiklik gösteriyordu. Bu dilin ilk taslağında hiçbir grafik imkânı öngörülmemiştir.

Bütün bu anlatılanlardan sonra, sizin için hangi dilin en uygun dil olduğuna bilmem karar verebildiniz mi? Eğer veremediyseniz cesaretiniz kırılmasın. En iyisi, kişinin bunu bizzat denemesi ve mizacına en uygun düşen dili sonunda kendi kendisinin seçmesidir. Programlama dillerinin seçimiyle araba ya da bilgisayar seçimi arasında pek fazla bir fark yoktur.

LOGO

AFİF SAY

"Bilgisayar sizi değil, siz onu programlayın."

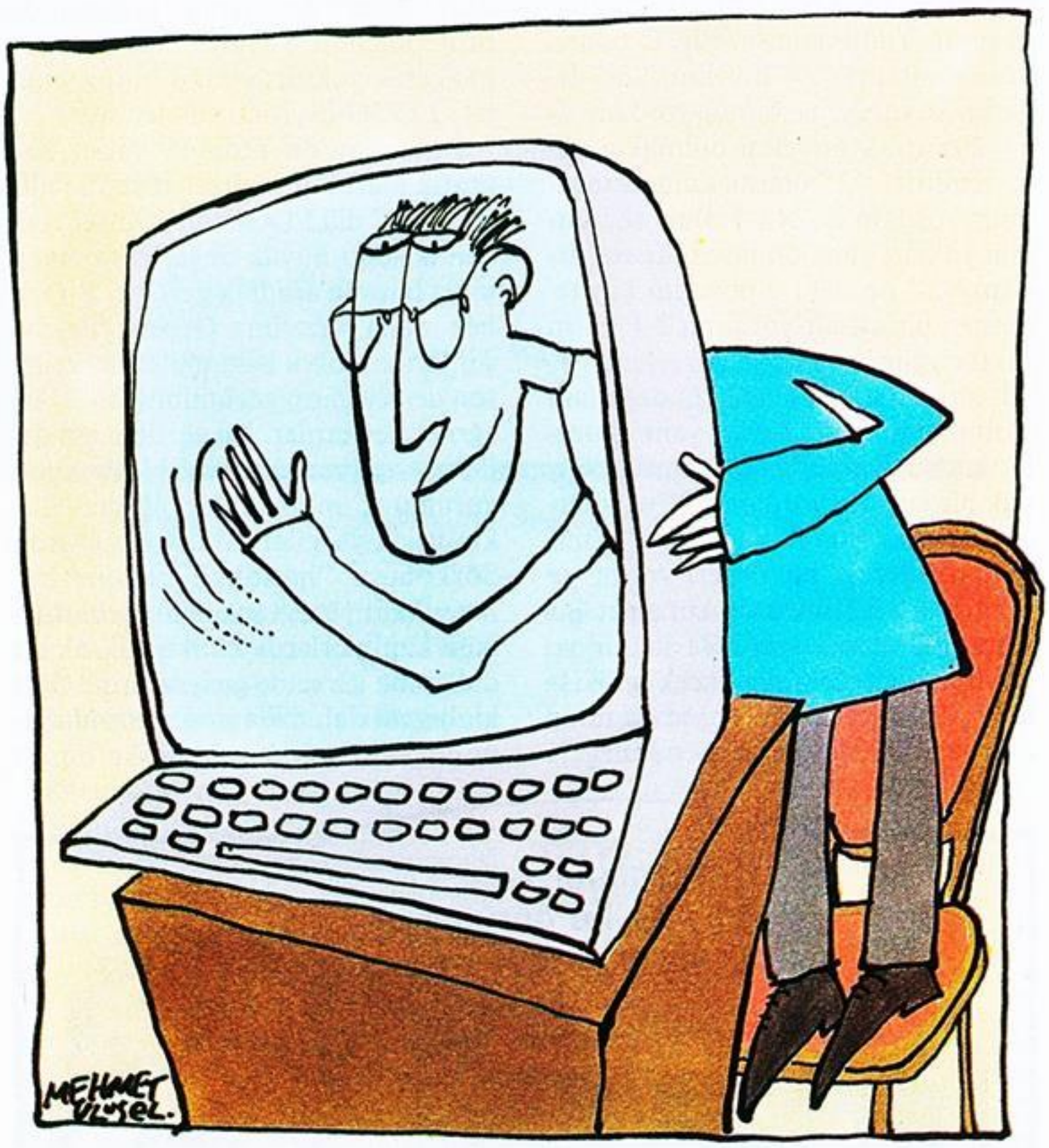
Bu sayıda tanıtacağımız dillerden birisi olan LOGO'nun felsefesi yukarıdaki cümleyle özetlenebilir. Bu da nerden çıktı demeyin, yoksa bu vakte kadar bilgisayarı programladığınızı mı sanıyordunuz? Evet görünüşe göre öyle, ama ya size yaptıklarınızın belli işlemleri yaptırabilmek için hep onun suyuna gitmek olduğunu istediği şeyleri hep onun istediği biçimlerde verdiğinizi söylersem? Yani aslında onun sizi programladığını...

İşte size bilgisayarınızı gerçek anlamda 'programlamanızı', onunla onun istediği gibi değil, kendi istediğiniz gibi iletişim kurmanızı sağlayacak bir dil: LOGO.

LOGO öncelikle bir eğitim dili olarak tasarlanmış (lütfen eğitimle öğretimi karıştırmayalım), bu nedenle özellikle çocuklar düşünülmüş, ama yediden yetmişe herkese yönelik olduğu görülmüş daha sonra. Çocukların genel yapıyı bir haftada kavradığı söyleniyor, büyükler için bu süre biraz daha uzunmuş.

LOGO yaklaşık onbeş yıl kadar önce İsveçli psikolog Jean Piaget'in öğretileri temel alınarak Seymour Papert'in öncülüğünde ABD'de geliştirildi.

Piaget'ye göre, "...insan çevresini yönlendirerek" öğrenir. Dili, sözcükleri yönlendirerek öğrenmeye başlar ve aşama aşama geliştirir. Bebek önce anlamsız sesler çıkarır, sonra sözcükleri kullanır ve daha ileri aşamada cümleler halinde birleştirir. Çocuk soyut bir kavram olan dili somut bir amaca ulaşmak için kullanır. Örneğin "su" sözcüğüyle su istediğini belirtip gereksiniminin karşılanmasını sağlar. Çevresiyle kurduğu bu iletişimi geliştirdiği zaman dilin gramatik yapısını irdelemeye başlar. Bu şekilde, yeni durumlarla karşılaştığında dili yeniden oluşturur.



İşte LOGO dili bu aşamalardan geçilerek öğrenilir ve geliştirilir. Önce grafikler öğrenilir -bebeğin konuşması-.

Grafikler somut sonuca karşılık düşerler. LOGO'da 'turtle' denen üçgen şeklinde bir yaratık istenildiği gibi yönlendirilerek çizgi çizmesi sağlanır. FORWARD (ileri git), RIGHT (sağa dön) komutları ile ekranda çizim elde edilebilir. Verilen bu komutlarla somut sonuçlara ulaşma başarısı arttıkça LOGO'nun dil yapısı da öğrenilir.

Sonra bu komutlar biraraya getirilerek daha karmaşık bir sonuca yönelik program parçaları elde edilir. Çocuk artık cümleler kurmaya başlamıştır. Artık soyut kavramlar geliştirerek yeni LOGO sözcükleri yaratılabilir.

Yukarıda açıklanan bu aşamaları

örneklersek:
Basit bir üçgen çizimini

```
TO UCGEN
FORWARD 50
RIGHT 120
FORWARD 50
RIGHT 120
FORDWARD 50
END
```

komutlarıyla yapabiliriz.

Şimdi daha üst bir kavrama geçelim:

```
TO HAC
UCGEN RIGHT 90
UCGEN RIGHT 90
UCGEN RIGHT 90
                3
UCGEN RIGHT 90
END
```

Görüldüğü gibi daha önce verilen üçgen kavramı, bir üst işlemde haç oluşturmada kullanıldı, yani sözcükler cümle içinde değerlendirildi. Basit kavramlardan yola çıkılarak karmaşık düşünceler ifade edildi.

LOGO'da turtle grafikleri önemlidir, ama yalnızca başlangıç için. LOGO eşittir çizim ve grafik denemez. Bu dilin asıl önemi semboller üzerinde işlem yapmamızı sağlaması (symbol processing), yani düşünceleri somut nesneler gibi işleyebilmesidir.

Konuyu bir örnekle açarsak: 'Kuş' sözcüğü bize kanatları olan, uçan, yumurtlayan canlı varlıkları anlatan bir semboldür. İnsan beyni kuşları bu sembole bağlı özelliklerle algılar. Martıda bir kuştur; yani kuşların bütün özelliklerine sahiptir. Ayrıca martının su kenarında yaşama, suda batmadan durabilme gibi kendine özgü nitelikleri de vardır. Sonuçta bütün martılar kuştur, ama bütün kuşlar martı değildir yargısı insan mantığı için son derece basit bir işlemdir. İnsanlar sembolleri doğal olarak değerlendirir; nesnelere özellikler bağlar ve diğer şeylerin aynı sınıfa dahil olup olmadığını yargılar. Bunu BASIC dilinde yapabilmek için çok karmaşık matematiksel algoritmalar kullanmak gerekmektedir. Oysa LOGO'da yöntem insandakine çok yakındır. Kuşun özelliklerinin listesi yapılır ve bu özellikler değerlendirilerek diğer nesnelerin kuşlarla olan ilişkisini yargılayabiliriz. Şimdi bütün bunların LOGO'ca nasıl ifade edildiğini görelim:

```
MAKE "KUS (LIST
(KANATLARI VAR) (UCAR)
(YUMURTLAR) (CANLIDIR))
MAKE "MARTI (LIST
(KANATLARI VAR) (UCAR)
(YUMRUKLAR) (CANLIDIR)
(SU KENARINDA YASAR)
(SUDA BATMADAN DURUR)

TO YARGILAMA
MAKE "SAYI COUNT": KUS
IF: SAYI=O THEN PRINT (MAR-
TI BIR KUSTUR)
TEST MEMBER? FIRST: KUS:
MARTI
IFFALSE THEN PRINT (MARTI
BIR KUS DEGILDIR)
MAKE "KIS BUTFIRST": KUS
YARGILAMA
END
```

Görüldüğü gibi ilk bölümde her iki kavramın da özelliklerini tanımladık, sonra kuşun özelliklerini martınınkilerle teker teker karşılaştırdık. (Bunu BASIC'le deneyin bakalım.) Eeee, bütün bunlar neye yarar, biz doğal olarak bu işlemleri yapıyoruz zaten, bilgisayarın yapmasına ne gerek var diye düşünebilirsiniz. Haksız da sayılmazsınız, ama düşünün bir kez, ona bizim gibi düşünmeyi daha yeni öğrettik. Hele bunu biraz daha karmaşıklaştırır, daha geliştirirsek bizimle fikir tartışmasına girmesi bile yakındır. Yani yapay zekaya sahip olması.

Neyse daha fazla uzatmadan son sözlerimi özetleyeyim: LOGO sorularımızı doğal yollarla ifade edebilmemizi sağlar. Bunun ötesinde LOGO problem çözmede bir yaklaşım sağlar. Yaratıcı yolla problem kurma ve çözme yollarını öğretir. Bu öğrendiğimizi yalnızca bilgisayarla olan ilişkimizde değil, yaşamın her alanında kullanabiliriz. Kısacası LOGO ile düşünmeyi öğrenebiliriz. □

LOGO'nun C-64 için Türkçe uyarlaması TELETEKNİK tarafından yakında çıkarılıyor... Artık 65 ile ana dilimizle çalışacağız.

LOGO projelerinizi bekliyoruz...

Başvuru: Afif SAY - Teleteknik A.Ş.

BİLTEK
BİLGİSAYAR VE TEKNOLOJİ TİCARET

Commodore 64'te yeni boyutlar:

YİNE BİLGİSAYAR DÜNYASININ TÜM YENİLİKLERİ VE
TAMİR SERVİSİMİZLE HİZMETİNİZDEYİZ
GÜNÜN YENİLİKLERİNDEN SEÇMELER

DOLPHIN DOS:
22 kez hızlı yükleme, 749 block free monitor üzerinde, kasete kayıt imkanı.

SPEEDOS PLUS:
17 kez hızlı yükleme. Özel modu ile 220 block 4 saniyede yüklenir.

64'ER DOS V3:
Diskten 17 kez hızlı yükleme.

EPROM PROGRAMLAYICI II:
2732-2764-27128-27256 tip epromları programlar (2764'8 saniyede)

802 ROM II:
802 printeri 801-803 ve epson haline getirir. (Printshop, print master v.s. MPS 802 ile çalışır.)

MODÜL KART:
Programlarınız kart üzerine hazırlanır.

DESTEK 64:
Modulle yardımcı programlar dizisi.

EXPERT KARTUŞ:
Programları tek parça halinde kırar, kasete istediğimiz turbo ile atma imkanı sağlar.

PRINTER INTERFACE:
Her türlü CENTRONICS I.E.E.E.C-488/RS232 printer C64'e adapte edilir.

SES DIGITALISER:
Bilgisayarınızı kendi sesinizle konuşturmayı, şarkı söyletmeyi hiç denediniz mi? Ses digitaliser'iniz için bizi arayın.

PROGRAM SERVİSİ:
Her türlü geniş kapsamlı iş, eğitim, oyun programları ile hizmetinizdeyiz. 1987 oyunları seçmelerinden derlenen paketler. Oyun paketlerimizde kafa ayar programı ücretsizdir.
Bir kasette 20 oyun 3000 TL
Bir kasette 40 oyun 5000 TL
Günün tek oyunları 1500 TL
Siparişleriniz ödemeli gönderilir.

THE ICE MACHINE:
Program kırma, kasete ve diskete atma, en uzun parçalı programları kasete 45 turda atma, disketten 6 kez hızlı yükleme FREEZE FRAME'lerden üstün özellikler.

İmalatlarımız ve programlarımız hakkında ayrıntılı bilgileri içeren broşürümüzü isteyiniz.

EBUZİYA CAD. MARMARA ÇARŞİSİ NO: 105 BAKIRKÖY-İST. TEL: 583 14 90

Commodore

BİLGİSAYARLARIN DÜNYASINDA YAŞAMAYA HAZIR MISINIZ?

▶ *Bilgisayar alacaksınız! Hangi bilgisayar neyi, ne kadar yapar? Karşılaştırmak, en uygun olanı seçmek için bilmeniz gereken her şey BİLGİSAYAR ANSİKLOPEDİSİ'nde var.*

▶ *Bilgisayarınız var! En iyi nasıl kullanacaksınız? Daha iyi nasıl yararlanacaksınız? Temel bilgilerden başlayarak öğrenmek, öğrendiklerinizi uygulamak için BİLGİSAYAR ANSİKLOPEDİSİ okuyun.*

▶ *Bilgisayarla çalışıyorsunuz! Daha iyi ve daha etkin programları nasıl yazacaksınız, işinize nasıl uyarlayacaksınız? Basic, Fortran, Cobol, Pascal, Assembler veya Logo dillerini öğrenmek için BİLGİSAYAR ANSİKLOPEDİSİ alın.*

▶ *Gelecekle ilgileniyorsunuz! Bilgisayar çağında ne olacak? İnsanlar nasıl yaşayacak, ne yapacak? Bilgisayarların toplumsal düzenine hazırlanmak için BİLGİSAYAR ANSİKLOPEDİSİ'ne başvurun.*

7 cilt, 2008 sayfa, sayısız şema ve tablo, ofset baskı, 1. hamur kâğıt, lüks cilt.

Kullanılan terimlerin Almanca, İngilizce, Fransızca karşılıkları.

Commodore Kulüp üyelerine % 10 indirimlidir

7 ciltte 6 Bilgisayar dili öğretiyoruz.

© Armando Curcio Editore, İletişim Yayınları.

T.C. Milli Eğitim, Gençlik ve Spor Bakanlığı tarafından okullara tavsiye edilmiştir.

		Peşin	TAKSİTLİ SATIŞ 4 ay	8 ay
BİLGİSAYAR ANSİKLOPEDİSİ (7 CİLT)	TOPLAM FİYAT	55.300	62.250	69.300
	PEŞİNAT		18.250	15.700
	TAKSİT		11.000×4	6.700×8

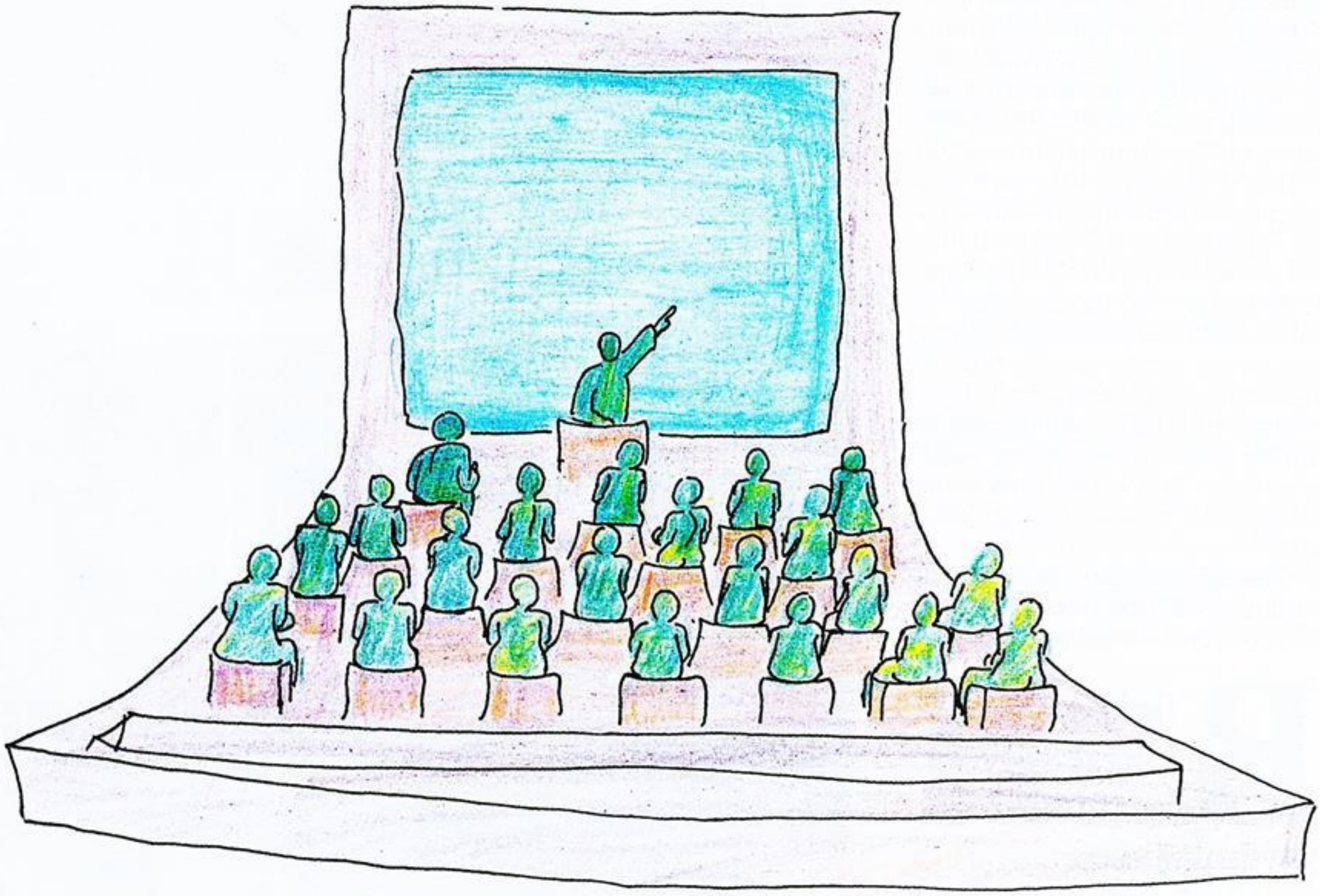
Adım: .....
Soyadım: .....
Adresim: .....
Telefonum: .....

Yazışma Adresi: İletişim Pazarlama-Klodfarer Cad. No: 7 34400 Cağaloğlu/İST.
Tlf: 512 17 32-33

BİLGİSAYAR
ÇAĞDAŞ TEKNOLOJİ
ANSİKLOPEDİSİ

İletişim Çağında Bilgisayar;
Bilgisayar Çağında İletişim

Yayınları • İletişim Yayınları • İletişim Yayınları • İletişim Yayınları • İletişim Yayınları • İletişim Yayınl



Sonunda Tartışıldı:

Bilgisayar ve Toplum

Bilgisayar ve Toplum konulu panel, 10 Nisan 1987 Cuma günü, Cumhuriyet Kitap Klübü tarafından FM Kültür Merkezi'nde gerçekleştirilen kitap fuarı kapsamında yapıldı. Panele Doç. Dr. Eşref Adalı, Ram Dergisi Genel Yayın Yönetmeni Bülent Aydemir ve Commodore Dergisi Yayın Yönetmeni İskender Savaşır konuşmacı olarak katıldılar.

Hayatımızda giderek daha geniş bir yer tutan bilgisayar özellikle iş dünyasında çeşitli düzeylerde tartışılıyor olmakla birlikte, olayın toplumsal boyutu henüz çok sınırlı bir platformda tartışma konusu ediliyor. Bilgisayar olayını bir buzdağına benzeten görüşten yola çıkarak, suyun altındaki kısmı da gözardı etmemek, olayın sağlıklı değerlendirilmesi ve geleceği açısından önem taşıyor. İnsanlığın tarihinde önemli dönüm noktalarından birini teşkil eden bu buluşun, toplumsal yansımaları ve karşılıklı etkileşimin konu edildiği böylesi tartışmalardan biri de Bilgisayar ve Toplum konulu paneldi.

Bülent Aydemir, ilk tur konuşmalarını şu sözlerle açtı: Bilgisayar olayının genellikle bize yansıdığı gibi teknolojik boyutundan başka sosyal boyutunun da olduğu unutulmamalıdır. Türkiye açısından olaya bakıldığında, "Bilgisayar ile, kaçırılan Birinci Teknolojik Devrim'den sonra İkinci Teknolojik Devrimi'ni yakalamak olası mı?", "Türkiye bilgisayar ile bu anlamda bir atak yapabilir mi?" sorularını sorabiliriz.

Bu sorular doğrultusunda bilgisayarın teknolojik gelişiminden başlarsak: Dünyada bilgisayarın teknolojik gelişimi nasıl olmuştur ve ileriye yönelik gelişim olanakları nelerdir?

Eşref Adalı: Bugün kullanılmakta olan bilgisayar sözcüğü aslen yanlış bir sözcük. Fakat artık bu sözcüğün değiştirilmesi pek mümkün gözükmüyor. Bilgisayarın gelişimini 1880'lerden başlatabiliriz. Bugün artık çok ilkel kalan delikli kartlarla. Daha sonra 2. Dünya Savaşı sırasında askerî amaçla geliştirilen bilgisayarlar var ki, bunlar aralarında dolaşılacak kadar büyük aletler. Üstelik kullanım sürelerinin ancak % 10'unda çalışabiliyorlardı. Geri kalan

zamanda boş durmak zorundaydılar. Nihayet 1950 yıllarında transistörün keşfedilmesi ile bilgisayar kiralanabilir ve satın alınabilir oldu. Türkiye'ye ilk bilgisayar ise Marshall Yardımı ile birlikte Karayolları'na geldi.

1971'de bilgisayar alanında önemli gelişmeler kaydedildi. Bu tarihe kadar bir bilgisayar bir buzdolabı büyüklüğünde iken, 1971'deki gelişmeler ile bu boyutlar büyük ölçüde küçüldü. Bilgisayar alanındaki gelişmelere birkaç örnek verecek olursak: 1971'den önce 10.000 dolar ile bir ev bilgisayarının bellek kapasitesine sahip bir bilgisayar alınırken, bugün aynı fiyata aynı bellek kapasitesine sahip 3 adet kişisel bilgisayar alınabilir.

Geleceğe baktığımızda ise bilgisayarın çok daha gelişeceği ve küçüleceği görülüyor. Bu gelişmelerle birlik-

Bilgisayarların giderek küçülmesi ve yaygınlaşması toplumların gelişiminde bilgisayarların rolünü ön plana çıkardı.

te elbette yaygınlığı da artacak. Çağımızda artık bilgisayarın kullanılmadığı bir toplum düşünmek olanaksız. Her dalda olduğu gibi bilgisayar da, kendi alanında çeşitli uzmanlık dallarını gerekli kılacak. Elbette herkesin uzman olması beklenemez. Fakat günümüzde her ferdin en azından bilgisayar kullanımı hakkında asgari bilgilere sahip olması bir zorunluluk olarak kendini belli ediyor.

Bülent Aydemir: Bugün bilgisayar sektöründe belli teknolojileri üreten, ve pazarlayan kuruluşlar var. Bilgisayar teknolojisindeki gelişmeler, dünyada A.B.D. ve Japonya'nın önderliğinde gerçekleştiriliyor.

Dünya piyasasında gelişkin teknolojilere sahip ülkeler bu teknolojileri kullanarak bilgisayar üretiyorlar. Bugün bilgisayar dünyasında belli teknolojileri üreten ve pazarlayan ülkelerin başında A.B.D. ve Japonya geliyor. Türkiye'nin bu ülkelerle bu anlamda rekabet etmesi sözkonusu olamaz. Bizde üretim montaj aşamasında. Teknolojik Devrimi gerçekleştirmiş ülkelerin ürettiği entegre devreleri ve mikro-işlemcileri alıp, montaj yapma aşamasında bir üretim gerçekleştiriyoruz.

RAM Dergisi'nin düzenlediği Bilgisayar ve Toplum panelinde konuşmacılar: (Soldan sağa)

Bununla birlikte montaj sanayisi ve buna bağlı yan sanayi kollarının gelişmesi de Türkiye açısından önemli ve üzerinde durulması gereken bir konu. Gelişmiş bir montaj sanayisinin yaratacağı önemli avantajların başında, yan sanayilerle birlikte önemli bir istihdam potansiyeli, Türkiye için hayati önem taşımaktadır.

Dünyada montajcı ülkeler olarak bilinen Kore ve Taiwan gibi ülkelerin ürettiği, kalite açısından yetersiz malların iç piyasada yaygınlaşmamasına dikkat edilmesi gerekiyor. Şu anda bilgisayar ithali ile ilgili yeni tasarılar var. Daha önce Kore ve Taiwan'dan mal getirip satan ve kısa vadede büyük kâr elde edip ortadan yok olan, ne idüğü belirsiz şirketlerin tüketiciyi riske eden bu tür faaliyetleri de bu yeni tasarı ile önlenebilecek.

Bilgisayar teknolojisinin ülkede yaygın olarak kullanılması hizmet ve destek veren tek bir kuruluşun ya da birkaç kuruluşun varlığı ile mümkün değil. Bunun için öncelikle tüketicinin bilinçlendirilmesi ve sağlıklı bir yaygınlığın gerçekleştirilmesi gerekmektedir. Bu konuda devletin desteğine de özellikle ihtiyaç vardır. Devlet, örneğin ikinci kanal gibi elinde bulunan olanakları bilinçli bir tüketici kitlesinin yaratılması için kullanabilir.

Bilgisayar tüketiminde asıl sorun, bilgisayar aldıktan sonra başlıyor. Bilgisayar, olayın sadece bir kısmı. Yarısı denebilir. Diğer yarısı ise prog-

Eşref Adalı, Bülent Aydemir ve İskender Savaşır.

ram. Programcılığın önemi henüz kavranmış değil. Bugün Türkiye'nin entegre devre üretimine girişmesi son derece gelişkin teknolojilerle rekabete girmesi demek olacakken, herhangi bir yatırım gerektirmeyen programların dünya çapında kullanılması, Türkiye'nin bu teknolojiyi bir yerlerden yakalaması demek olacaktır. Fakat bunun bilincine henüz tam olarak varılmadığından ve bu yönde ciddi çalışmalara henüz girişilmediğinden bu şimdilik uzak bir olasılık.

İskender Savaşır: Korkarım, diğer konuşmacıların temkinli iyimserliği ile karşılaştırıldığında bile, benim konuşmam fazlasıyla karamsar kaçacak.

Şimdi Bilgisayar Devrimi dediğimizde kasdettiğimiz şeylerden biri de, bilgi-işlem teknolojisinin hâkim teknoloji, egemen teknoloji ya da Nora Şeni'nin kontrol teknolojisi dediği şey haline gelmesi. Ne demek, peki, kontrol teknolojisi? Eğer herhangi bir alandaki üretkenlik, verim ve teknolojik yenilikler ekonominin diğer bütün alanlarına da yansıyorsa, sözkonusu alana hâkim teknoloji deniyor. Bir örnekle açıklamaya çalışayım: Bugün eğer ömür çıkarma teknolojisinde teknolojik bir sıçrama görülürse, bu, kuşkusuz önemli olur. Örneğin ne bileyim ben, kömürün fiyatı ucuzlar, ilişkili bazı yan dallarda etkileri hissedilir. Ama bütün bu etkiler lokal kalmaya mahkûmdurlar. Çünkü kömürün elde edilmesinde kullanılan teknoloji bugün hâkim teknoloji değildir. Oysa bilgi-işlem teknolojisi bugün artık hâkim teknoloji haline gelmiştir. Üstelik bilgi-işlem teknolojisinin tarihte hâkim olmuş diğer teknolojilerden de bir farkı var. Daha öncekiler ekonominin "sanayii" diye adlandırdığımız alanlarını etkiliyordu. Oysa bilgi-işlem hizmet sektörü diye adlandırdığımız alanları da, bir anlamda sanayileştirdi.

Bu kadar kritik bir noktada duran bir teknolojinin, bir sektörün yapılanması, kurumlaşması üzerinde de durmak gerekir. Çünkü hayatın iktisadî olmayan yönleri de, neredeyse hayatın tamamı bunlardan etkilenecektir.

Önce şunu vurgulamak gerekiyor: Tarihte hâkim teknoloji konumuna gelmiş teknolojilerin daima merkez ülkelerde, birinci dünya ülkelerinde yoğunlaştığını görüyoruz. Örneğin, Türkiye'de demir-çelik sanayiinin kurulması, ancak bu teknoloji artık hâkim teknoloji olma konumunu yitirdikten sonra mümkün olabilmiştir.

Hakim teknolojiye sahip gelişmiş ülkeler bilgisayar alanında da neyin ne kadar üretileceğini belirliyorlar.

İkinci olarak neyin ne kadar tüketileceği ile hâkim teknoloji arasındaki ilişkiye değinmek gerekiyor. Bundan önceki dönemde hakim teknoloji yassı çelik üretimi idi. Yassı çelik üretiminin kârlı bir biçimde sürebilmesi ise, ancak yassı çelik ürünlerinin büyük ölçekte satılması ile mümkündür. Bu ise kitlesel tüketim demektir.

Neydi bu kitlesel olarak tüketilen yassı çelik ürünleri? Buzdobalı, çamaşır makinesi, fırın, kısaca "beyaz eşya" diye andığımız şeyler.

Şimdi söyleyeceklerim ancak izlenim düzeyinde şeyler. Bu iddiaların doğruluğunu sınamak için gereken verilere nasıl ulaşılabileceğini dahi bilemiyorum. Ama Türkiye'de beyaz eşya imalatı başladıktan sonraki beş yıldaki, sözgelimi buzdolabı satış artış hızı ile bilgisayar ithalatı başladıktan sonraki bilgisayar satış artış hızı karşılaştırılırsa, bilgisayar satışlarının çok daha az olduğunu göreceğiz. Dediğim gibi, bu ancak izlenim düzeyin-

Commodore

de bir iddia. Ama eğer doğruysa ne anlama geliyor?

Kitlesel tüketim ancak geniş bir orta sınıfın varlığıyla mümkün olur - 1960'lar Türkiyesi'nde olduğu gibi-. Eğer bilgisayar bir lüks tüketim maddesi olarak kalacaksa... Ya da şöyle söyleyelim... Şimdiki iktisadi sistemin taşıyıcısı durumunda olan bilgisayarın kitlesel olarak tüketilmesi gerekmiyorsa ne olur? Bir kere alıştığımız orta sınıf iktisadî olarak işlevsizleşir, dolayısıyla kaybolur. Eşitsizlikler artar. Hem merkez ülkelerin kendi içindeki sınıfsal eşitsizlikler, hem de birinci dünya ile üçüncü dünya ülkeleri arasındaki eşitsizlikler.

Eşitsizlik ise hiçbir zaman *sadece* eşitsizlik değildir; dünyaların ayrışması anlamına gelir. Bütün bunların çeşitli toplumsal implikasyonları var. Örneğin, bu konularla ilgilenen bizim gibi burada toplanmış aydınlarla toplumun diğer kesimleri arasında, kurulan ilişkinin niteliği hakkında. Ama bunlara belki daha sonra tartışma bölümünde gireriz.

Bülent Aydemir: Bugünlerde gündemde olan ve üzerinde çokça konuşulan konulardan biri de, Avrupa Akademik Araştırma Ağı (EARN). Bu ağa dahil olmakla, Avrupa'da bilgisayar alanındaki pek çok yenilik ve kazanımdan da doğrudan yararlanabilmek olanağına kavuşuluyor.

Eşref Adalı: EARN biraz kurcalanırsa, ticarî bir firmanın reklamı olduğu anlaşılır. Avrupa'da buna benzer ağlar var. Amaç çeşitli kazanımların ortaklaşılması. Bundan çeşitli kazançlar sözkonusu elbette ama olaya çok idealist yaklaşmamak gerekiyor. Örneğin bilgisayar alanında en son yeniliklere, özel çalışmalara bu yolla ulaşılacağı gibi bir düşünceye kapılmamak gerekiyor.

Türkiye'de bilgisayarın yapılıp yapılamayacağı tartışmasına gelince, bunu tartışmaya hiç gerek yok. Türkiye'de yıllardır bilgisayar yapılıyor. Asıl "satılıyor mu?" diye sormak gerekiyor. Türkiye'de bilgisayar bilgisizlik ya da başka bir yetersizlikten değil satılmadığından, yani çok yüksek maliyet sözkonusu olduğundan üretilmemektedir.

Bilgisayar çok hızla gelişen bir alet. Bu hıza erişmek ise yine çok hızlı ve gelişkin bir teknolojiyi gerektiriyor. Donanıma yatırım yapmanın cazibesi yok ama fason üretimin cazibesi var. Ya da belli alanlarda uzmanlaşmak gerekiyor. Kişisel bilgisayar sözkonusu olduğunda, bugün yılda bir milyon tane satamıyorsanız, üretmek anlamsızdır. Ama aynı şey daha büyük sistemler için geçerli değil. Bir Norveç firması birkaç akademisyenin yaptığı bir bilgisayarı dışarı satar. Yılda yirmi tane satar ama satar. Bu Norveç açısından avantajdır.

Ülkemizde bilgisayar alanında montaj üretiminin ötesine geçmek zor. Ama programcılıkta bir şeyler yapmak pekâlâ mümkün.

Türkiye beyin açısından çok verimli ve zengin bir ülke ve çok iyi programlar yazılabilir. Ama dünya ile iletişimimiz sağlıklı olmadığından, bu kaynaklar da sağlıklı kullanılamıyor.

Türkiye'de yazılım için yatırım yapılabilir. Dünyada bilgisayar sektörüne bakıldığında, bu sektördeki harcamaların sadece % 10'u donanım alanındadır. Asıl önemli olan yazılımdır ve Türkiye'de de buna önem verilmelidir.

Bülent Aydemir: Türkiye'de bilgisayar olayına çeşitli sektörler itibarıyla baktığımızda, örneğin turizm kuruluşlarının bilgisayar sistemleri kurduğunu görüyoruz. Tek tek firmaların bu şekilde tek tek sistemlere yatırım yapmaları gereksiz olduğu gibi sektörel açıdan merkezîliği de olumsuz yönde etkilemektedir.

Yine, devlet sektöründe bilgisayar alımı ile ilgili ihalelere sık sık rastlamaktayız. Bu alımlar iki amaç için gerçekleştirilmekte: 1) Bilgisayar programcılığı için bilgisayar kullanılması, 2) Okullarda eğitimin bilgisayar ile yapılması.

Özellikle eğitim konusu üzerinde durmak istiyorum. M.E.B. bu konuda belli bir netliğe ulaşmalıdır. Ne yapılmak isteniyor? Bunun için kalifiye eleman var mı? Örneğin bir BASIC eğitimi sözkonusu ise, bu ilk, orta ve lise seviyesinde nasıl olacaktır? Okullarda programcılık ne şekilde yerleştirilecektir? Bilgisayar bir kitap ya da defter olarak kullanılacaksa, bu yatırımlar ne ölçüde işe yarayacaktır? Bugün bilgisayarın girdiği okullarda bilgisayar bir test aracı olarak kullanılıyor. Oysa öğrenci, konuyu grafik şekillerle anlatabilmeli, bilgisayarın çeşitli olanaklarını kullanabilmelidir. Öğrenci karşılıklı etkileşim ile canlı bir eğitim aracına kavuşmalıdır. Bu ise, özel olarak yazılmış birtakım programların varlığını gerektirmektedir. Piyasada bunlara uygun program bulunmuyor. Dışardan böyle programların getirilmesi üzerinde ise fazla durulmuyor.

Türkiye'de mevcut eğitim sistemi ortada iken, bilgisayar kullanımı ile amaçlanan nedir, nereye varılmak isteniyor bu pek net değil. Bununla birlikte bilgisayarın eğitim alanına sokulması salt bilgisayar alımı olarak kalacaksa, mevcut koşulların farklılaşması beklenemez.

İskender Savaşır: Yazılım konusunun önemine ve bu konuda bir avantajımız olduğuna ben de katılıyorum. Ama maalesef bu konuda da karamsarım.

Bakın, tarihte tüm teknolojik sıçramaların, o dönemin en az merkezî yapıları tarafından gerçekleştirilmiş olduğunu görüyoruz. Bu 15. yüzyıl İtalya'sı için de böyledir, 18. yüzyıl sonu İngiltere'si için de, kişisel bilgisayar devrimi için de. Bu son örnekte gayrı-merkezîlik kısmen sunî olarak yaratılmıştır. Kişisel bilgisayar devrimini gerçekleştiren tekeller değil, küçük firmalar olmuştur. IBM bu alana girmemiş, araştırma riskini küçük firmaların yüklenmesini beklemiştir. Sonunda piyasaya girmesininse teknolojik gelişme üzerinde olumsuz etkisi olduğu söylenebilir.

Demek ki, teknolojik yaratıcılıkla gayrı-merkezîlik arasında bir ilişki var. Oysa biz üniversitelerle ilgili olarak sürekli merkezî çözümlere yöneliyoruz. Bu tabiî gerekçesiz değil. Üniversiteler birer yaratıcılık odağı olduğu kadar, birer gayrı-memnuniyet odağıdırlar da. Dünyanın her yerinde bu böyledir.

Şimdi demin söylediğim eşitsizlikler arttıkça üniversitelerin bu vasfı daha da ön plana çıkacaktır. Dolayısıyla daha da merkezî çözümler gerekecektir. Bu ise üniversitelerin zeka ve yaratıcılık potansiyelini olumsuz yönde etkileyecektir.

Kısacası elimizdeki zeka potansiyelinin önümüzdeki yıllarda azalacağı kanısındayım. □

Commodore

SKY BİLGİSAYAR-ELEKTRONİK

DONANIMDA
YAZILIMDA
HİZMET

SKY şimdi de Nişantaşı'nda hizmetinizde

2 Mayıs'ta açılışımıza davetlisiniz

DOS SİSTEMLER:

SPEED DOS PLUS: *17 kez hızlı paralel dos. Özel format ile 35 kez hızlı.*
DOLPHINDOS 2.0: *25 kez hızlı paralel dos. 749 Block formatlama. Centronics çıkışı. Birçok yeni fonksiyon. (C64,C128,SX64,1541)*

KIRMA VE KOPYALAMA KARTUŞLARI:

EXPERT CARDRIDGE:*Programları tek parça ve açık olarak kırar.*
ICE MACHINE PLUS-SUPER PLUS: *Diske 2 parça, kasete 45 turda kopya.*
FINAL CARDRIDGE: *16 yeni komut + Fast load + Freezer + Hard copy + Monitör*
DOUBLE CARDRIDGE: *ICE MACHINE + FİNAL Bir arada.*

DİĞER ÜRÜNLERİMİZ:

C64 SES DIGITIZER:*C64'ü kendi sesinizle konuşturun.*
EPROM PROGRAMLAYICI: *27 serisi epromları programlar. Modül hazırlar.*
FAST LOAD MODUL – DESTEK TEYP MODUL

RS 232 INTERFACE – CENTRONICS INTERFACE

64 K SÜPER MODUL

- BİLGİSAYAR VE YAN BİRİMLERİ SATIŞI
- SPECTRUM VE AMSTRAD PROGRAMLARI
- SPECTRUM MULTIFACE TWO KOPYA KARTUŞU
- BİLGİSAYAR ONARIM SERVİSİMİZ HİZMETİNİZDEDİR
- COMMODORE 64 SÜPER 64 PAKET OYUNLARIMIZ ÇIKMIŞTIR
- HER TÜRLÜ ÖZEL DEVRE DİZAYNI
- ÜRÜNLERİMİZ 6 AY GARANTİLİDİR

HER TÜR SORUNLARINIZ İÇİN BİZİ ARAYINIZ

ÖDEMELİ GÖNDERİLİR

ADRES VE TELEFONLARIMIZ

- Ebu Ziya Cad. Özgiray Pasajı, 17 K.1
  Büro: 32 Bakırköy/İstanbul Tel: 583 54 03
- Vali Konağı Cad. Konak Pasajı Kat:1 No:38
  Nişantaşı/İstanbul Tel: 148 43 83

BROŞÜR İSTEME KUPONU:

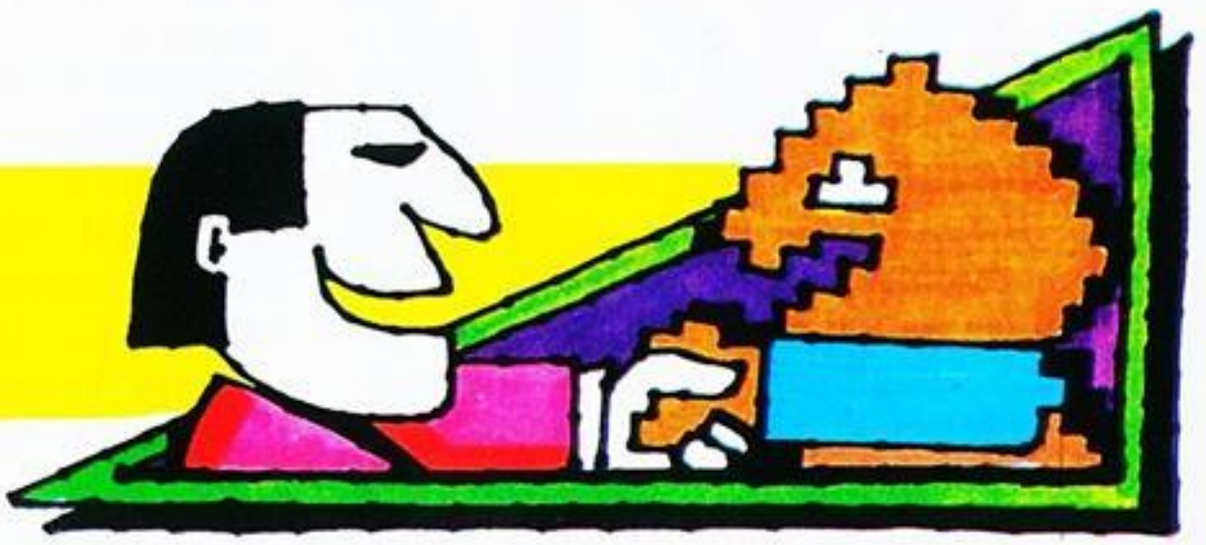
ADIM:
ADRESİM:
BİLGİSAYARIM:

☐ DISK ☐ TEYP

LÜTFEN AŞAĞIDA IŞARETLİ BROŞÜRLERİ GÖNDERİNİZ

☐ HARDWARE
☐ OYUN PROG.
☐ İŞ PROG.
☐ EĞİTİM PROG.

TANITIM



GUNSHIP

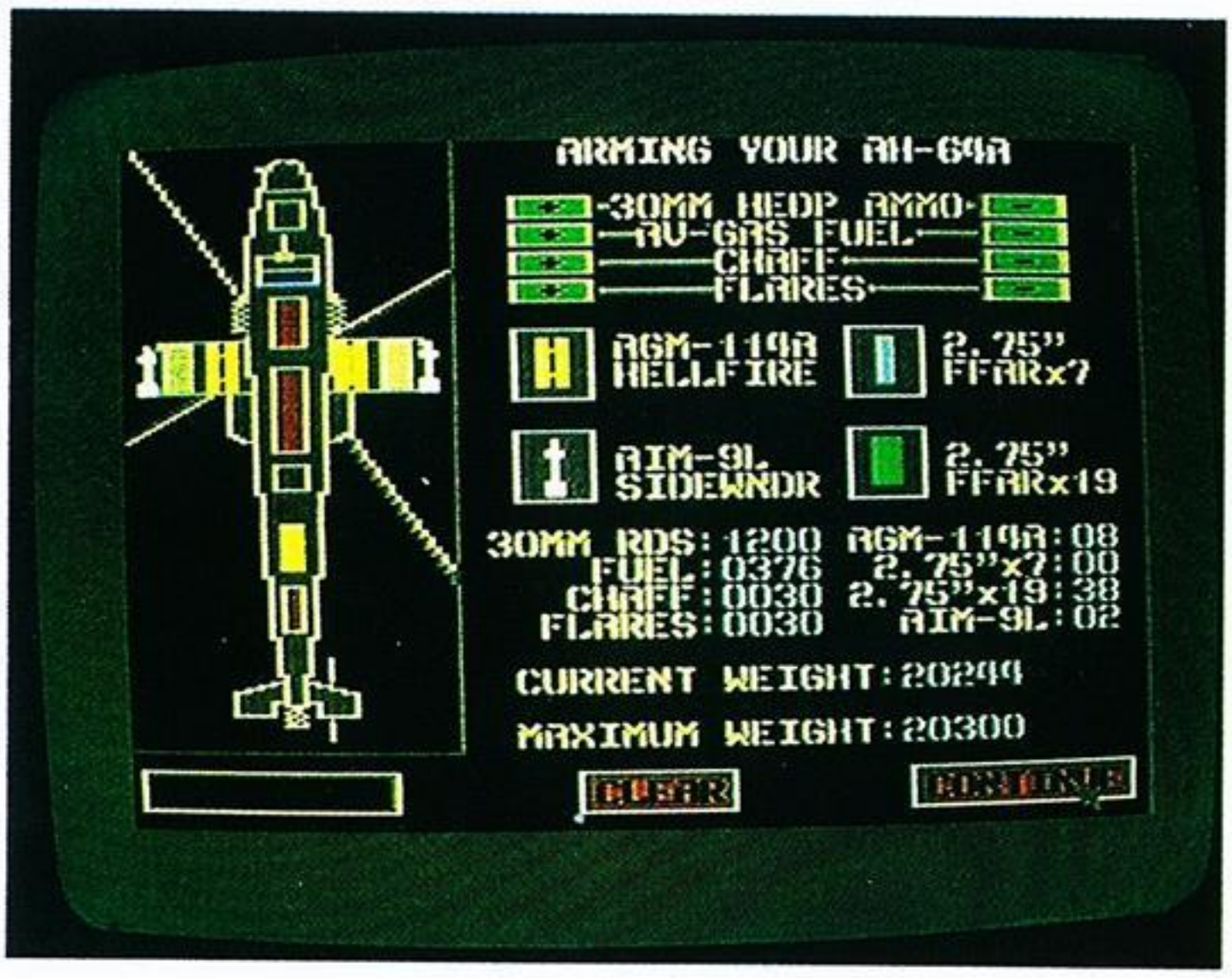
İnfiltrator gibi etkileyici grafikler içeren bir helikopter oyunundan sonra bu konuda görülecek bir şey kalmadığına inananlara ne kadar yanıldıklarını gösterecek bir program bu. Önceki programın aksine Gunship'te "simulation" elementi ağırlık taşıyor. Buna rağmen birçok "flight simulation" gibi grafiklerin kalitesi, programın karmaşıklığına kurban edilmemiş. Tabiî bunda programın disketten yükleme yapmasının da rolü büyük.

Program yüklendiğinde karşınıza gelen ilk ekranda o anki pilotun ismi, rütbesi, aldığı nişan ve madalyalar, puanı, görev yeri, görev türü, görevin gerçeklik oranı ve riski görülüyor. Ekranın üstünde görülen 5 seçenekten birini joystick yardımıyla yönettiğiniz küçük bir okla seçebilirsiniz. Bunlardan en soldaki (Pilot) ölü ve diri tüm pilotları incelemenizi sağlıyor. Bu pilotlardan KİA (görev sırasında öldü) ve MİA (görev sırasında kayıp) olanlar haricindekileri joystick ile seçebilirsiniz. İsterseniz Erase Pilot seçeneği ile bunlardan birini silerek kendi pilotunuzu da yaratabilirsiniz. Yeni pilotlar daima Sgt (çavuş) rütbesinden başlıyorlar, puan arttıkça rütbe de ilerliyor.

İlk ekranın ikinci seçeneği olan Duty size görev alanınızı seçme hakkını veriyor. Görev alanlarından ilki ABD içinde bir eğitim bölgesinde bulunuyor; bu görevi seçerseniz hiç vurulmama avantajını da kazanırsınız. Diğer görev alanları ise, zorluk derecesine göre sırayla Orta Amerika, Orta Doğu ve Batı Avrupa.

Style seçeneği ile görevin zorluk derecesi (regular en kolay, volunteer hazardous en zor) değiştirilebilir. Reality ise uçuş, iniş, hava durumu, düşmanın gücü gibi faktörlerin gerçeğe ne kadar yakın olacağını ayarlamanızı sağlıyor.

Pilot ve görev yeri seçiminiz sona erince Continue seçeneği ile brifing ekranına geçiliyor. Burada görevinizde yapmanız gerekenler, asıl ve yedek hedeflerinizin koordinatları gibi önemli bilgiler var. Gene joystick ile görebileceğiniz harita (Map) ve gizli servis raporu (İntelligence) ile görevinizi kolaylaştıracak ek bilgiler edinebilirsiniz. Görevden son anda vazgeçerseniz, Sick Call seçimiyle kaçma



şansınız var. Ancak bu yola pek sık başvurmasanız daha iyi olur, üstleriniz bundan hiç hoşlanmayacaktır.

Göreve başlamadan önce son bir defa her şeyi kontrol etmenizi öğütleyen Preflist Checklist ve bunu izleyen silahlanma ekranı göreceğiniz ekranların sonuncuları. Bunu da geçtikten sonra artık uçuşa başlayabilirsiniz.

Uçmak için ilk olarak 1 ve 2 tuşlarıyla her iki motoru çalıştırın; 3 tuşu ile ana rotoru döndürmeye başlayın ve F1/F3 ile gazı artırarak havalanın. Joystick ile hızlanma/yavaşlama (ileri/geri) ve sağa/sola yatarak dönüşleri yapabilirsiniz. Yere paralel durumdayken cursor tuşları ile sağa/sola (yatmadan) dönmeniz mümkün. F5/F7 ise gazı kısarak alçalmanızı ve yavaşlamanızı sağlıyor. Uçuş sırasında (pound) tuşuyla sol, DEL tuşuyla sağ taraftaki görüntüleri de görebilirsiniz. CLR tuşu tekrar ön görüntüyü verir. Commodore tuşuyla silahların durumu, soldaki SHIFT tuşuyla hasar durumu, Z tuşuyla da haritanın tamamı incelenebilir. Haritanın ufak bir kısmı zaten kokpit içinde devamlı olarak görülebiliyor.

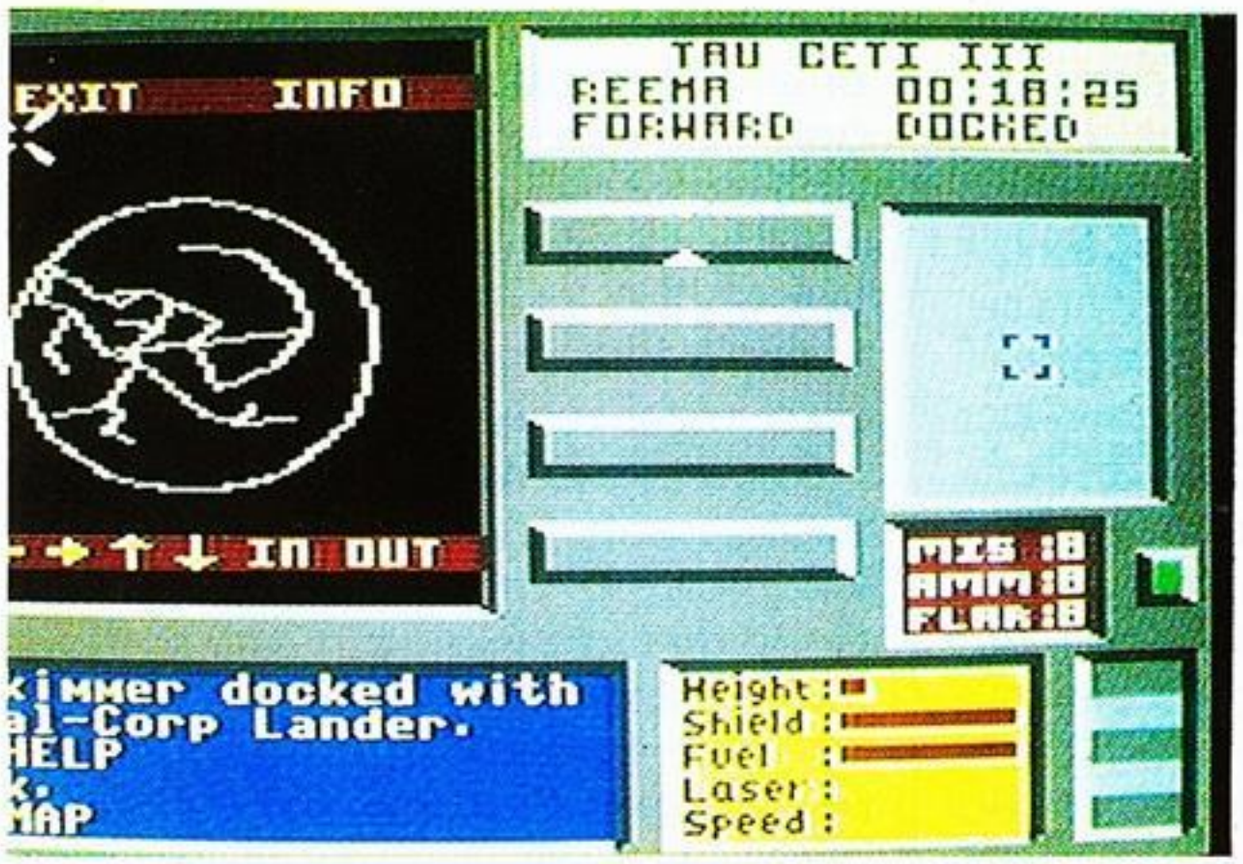
Uçuş sırasında herhangi bir hedefe yaklaşıldığında haritanın üzerinde TARGET uyarısı belirir. Space ile hedefin görüntüsü, cinsi ve uzaklığını alabilirsiniz. Nişangâh ile otomatik hedef kitleyicisi izlenerek, kitleyici renk değiştirince ateş edilir. Ancak bu arada hedef kitleyicinin dost/düşman ayırımı yapmadan her türlü yapı ve araca kitlendiğini de unutmayın. Eğer ateş açtığınız bir dost kuvvet ise hemen RADİO NSG ibaresiyle bir radyo mesajı gelecektir. Gene space tuşuyla bu mesajı okuyabilirsiniz. Ateş açma her zamanki gibi joystick düğmesiyle yapılıyor. Silah seçiminde ise 4'ten 9'a kadar rakamlar kullanılıyor. Bunlardan 7 tuşu ağır makineli tüfek seçmenizi sağlıyor. Tavsiyemiz, kullanımı en kolay olan bu silahı seçmeniz. Kokpitte göreceğiniz kadranlara gelince; bunlardan en sağdaki radar, onun solunda altta yükselme/alçalma açısı, hemen üstünde altimetre (yükseklik), ortada hedef ve harita ekranı, sol altta yere göre eğim (banking), üstünde hız göstergesi bulunuyor. Silahlar ise harita ekranının hemen altında. En soldaki küçük kadran pusula, onun sağındaki dikey göstergelerden F ile işaretlenen yakıt, T motor gücü (thrust), R ise rotor hızını gösteriyor. En üsteki harfler ise hasar gören yerleri belirtiyor, hasar arttıkça bunların renkleri değişiyor.

Eh bunca kontrolu aklınızda tutabilirseniz, uçuşta pek zorluk çekeceğinizi sanmıyoruz. Gene de başlangıç için Trainer görevlerini seçmeniz yararlı olur. Bu arada eğer görev sırasında ölürseniz tekrar aynı pilotu kullanamayacağınızı da unutmayın. □

TAU CETI

2047 yılında yıldızlar arası yolculuğun keşfi ile insanlık için gelecek başka bir anlam kazandı. Uzay gemileri uzayda dolaşan hidrojen akıntılarından çektikleri hidrojeni yakıt olarak kullanarak uzayın derinliklerinde yolculuk yapmaya başladılar. Bundan üç yıl sonra kolonicilerin ilk dalgası dünyadan ayrılarak dört-G tipi yıldızlar olan ALFA CENTAURI, TAU CETI, VAN MAANEN'S STAR ve BETA HYDRI'ye doğru yola çıktılar. Böylece insanlığın yıldızları kolonileştirmeye başlaması gerçekleşmiş oldu.

Bu dört gezegen içinde TAU CETI Yaşam şartlarının en zor olduğu gezgendi, 90 yıl boyunca buraya yerleşmiş olan kolonistler güç şartlar altında gezegenin olumsuz koşullarına karşı savaştılar. En sonunda bu kişiler gezegende otuz



şehir kurup madencilik alanında ticarette ve robot teknolojisinde çok güçlü duruma geldiler. ENCKE vebası diye bilinen çok öldürücü ve bulaşıcı bir hastalık bütün gezegeni kasıp kavurmaya başlamadan çok kısa bir süre önce gezegendeki her şey yolunda gitmeye başlamıştı ama bu salgından gezegende yaşayanların hemen hemen hepsi öldü, kurtulabilen çok az sayıdaki insan ise gezegenin güvenliğini robot defans mekanizmasına bırakarak gezegenden ayrıldılar.

Dünyadaki doktorlar ve bakteriyologlar derhal bu ölümcül virüsü etkisiz hale getirmek için araştırmaya başladılar. Çalışmalara başladıktan iki sene sonra hastalığa çözüm bulundu ve böylece koloniler arasındaki bağlantı tekrar gerçekleştirildi. Geri kalan üç gezegendeki koloniler federasyonu tarafından TAU CETI'nin tekrar kolonileştirilmesi kararlaştırıldı, bundan birkaç ay sonrada bir uzay gemisi TAU CETI'ye doğru yola çıktı. Gemi gezegene vardığında gezegene bir meteor düşmüş olduğunu fakat gezegendeki hiçbir yapı veya tesisin zarar görmemiş olduğunu raporunu gönderdi. Böylece gezegene tekrar inme hazırlıkları başladı. Dünyadakiler her zaman olduğu gibi gezegene inişi monitörlerinden izlediler ama görüntü çok büyük bir statik engelleme tarafından bozuldu. Gezegene iniş yapan koloni grubunu uyarmak için artık çok geç kalınmıştı.

Gezegenin yüzeyine gönderilen insansız bir araçla gerçek anlaşıldı. Düşen meteor yüzünden güvenlik sistemi bozulmuş ve her şeyi düşman saymaya başlamıştı. Robot güvenlik sistemini elle kumanda etmek için planlar yapıldı ve bu iş için en uygun olarak gezegene silahla donatılmış bir uzay arabası gönderilmesine karar verildi. Bu aracın görevi merkezi şehir CENTRALIS'teki füzyon reaktörünü havaya uçurmaktı. Yedi yıllık bir araştırma, tartışma ve eğitim sonunda tek kişilik bir gemi sizin kumandanızda gezegenin atmosferine girdi.

Görevinizin başarısı tamamı ile sizin aracınızı kullanmaktaki kabiliyetinize bağlıdır. Küçük ama faydalı aracınız görevinizde size yardımcı olmak üzere birçok faydalı özellikle donatılmıştır: Dört yönde görüş imkanı, sekiz güdümlü misil, sekiz antimisil, sekiz FLARE. Göreviniz sırasında ihtiyacınız olacak bilgiler için standard bir bilgisayarda aracınıza yerleştirilmiş bulunmakta. Aracınız gezegen yüzeyinin hemen üstünde ilerlemekte fakat yerden yüksekliğini isteğinize göre belirli bir ölçüde değiştirebilirsiniz. Bir şehirden diğer bir şehire geçmek için JUMP PAD üzerine gelerek J tuşuna basmanız yeterli olacaktır. Bu şekilde gezegenin herhangi bir yerinde bulunan istediğiniz şehire gidebilirsiniz.

Oyunda kullanacağınız komut sayısı oldukça çoktur, oyun başlar başlamaz aracınızın bilgisayarına HELP yazarak mevcut olan bütün komutları öğrenebilirsiniz, aynı şekilde aracın bilgisayarından yararlanarak kullanmanız gerekli tuşları da öğrenebilirsiniz. Şimdi bu komutların bazılarını inceleyeceğiz ve ekrandaki bölümleri belirteceğiz.

Ekran üç ana bölüme ayrılmış durumda: aracın ekranı, bilgisayar ekranı ve bilgi gösterim bölgesi. Aracın ekranından dışarıda ne varsa onu görmektesiniz. Bilgisayar ekranında hem komut girebilirsiniz hem de bilgisayarın cevabını görebilirsiniz. Bilgi bölümü ise beş parçaya ayrılmış durumda: çevreyi gösteren bir radar, geçen oyun süresi, pusula, o an bulunduğunuz yer ve durumunuz. Bunlardan başka iki yön bu-

lucu var, bunlardan ilki içinde bulunduğunuz şehrin merkezini işaret ediyor diğeri ise size en yakın JUMP PAD'ı işaret ediyor.

Aracınız binalardan birine girdiğinde bilgisayarınızın kabul edeceği bazı kullanışlı fonksiyonları bulunmakta. MAP komutunu verdiğinizde karşınıza gezegenin bir haritası çıkacak, Joystickle oklara basarak haritanın istediğiniz bölümlerini görebilir, ZOOM komutu ile haritanın belirli bir bölümünü büyütebilirsiniz. İnceleyebileceğiniz 32 şehir bulunmakta. Ateş tuşuna bastığınızda size WHICH CITY (hangi şehir) sorusu sorulacak, istediğiniz şehrin ismini yazdığınızda şehir hakkında bilgi alabilirsiniz. PAD komutunu girdiğinizde karşınıza boş bir sayfa çıkacaktır, bu sayfaya sizin için önemli olan notları alabilirsiniz (oyun sırasında bunun ne kadar önemli olduğunu anlayacaksınız). NEW PAD komutu ile kendinize yeni bir sayfa açabilirsiniz. Oyunu SAVE edebilir veya eski bir oyunu LOAD edebilirsiniz. KEYS komutu ile kullanacağınız tuşları yeniden tanımlayabilirsiniz. Girdiğiniz binalarda geminiz tamir edilir ve yeniden donatılır.

Oyundaki başlıca amacınız, CENTRALIS şehrindeki ana merkezi reaktöre girmektir. TAU CETI'deki çeşitli şehirlerden topladığınız ROD'lar yardımı ile bir soğutma işlemi yapmanız gerekmekte. Bir binaya girdiğinizde LOOK komutu ile binanın içini görebilir ve binada işinize yarayacak eşyaları öğrenebilirsiniz.

Oyuna başladığınızda aracınız gezegene iniş yaptığınız GAL-CORP gemisinde bulunmaktadır. LAUNCH komutu ile bulunduğunuz binadan gezegenin yüzeyine çıkabilirsiniz. İniş yaptığınız şehir REEMA'dır. TAU CETI gezegeninin bir güneşi bulunmakta, güneş etrafı aydınlatırken binaların gölgeleri giderek değişir, güneş battığında ise etraf tamamen karanlığa büründüğü için aracınızın INFRA RED sistemini açmanız gerekir.

Şehirlerin bazıları robotlar tarafından yönetilen araçlarla, diğerleri ise lazer kuleleri ile kendilerini savunurlar. Saldırgan bir nesne ile karşılaştığınızda en akıllıcası elinizden geldiği kadar çabuk rakibinizi yok etmeye çalışmalısınız, eğer bunu başaramazsanız rakibiniz size ateş etmeye başlayacak böylecede aracınızın kalkanları zayıflayacaktır, eğer dikkat etmezseniz aracınızı kaybedersiniz.

Gezegende bulacağınız kırk parça ROD'u toplayıp bunlardan yirmi tane bütün ROD üretin, daha sonra CENTRALIS'teki ana reaktöre bunları yerleştirin. Bir binaya girdiğinizde, o binada ROD olup olmadığı size belirtilecek ve eğer ROD varsa bunlar otomatik olarak aracınıza yüklenecektir. ROD'ları birleştirme işlemini sadece bir binaya girdiğinizde yapabilirsiniz. RODS komutuna girdiğinizde karşınıza ROD ekranı çıkacak. ROD'ları birleştirme işlemi bir imleç yardımı ile yapılmakta, imleci bir parçanın üstüne götürün ve parçayı alın sonra altı pencereden birine yerleştirin. Daha sonra bu elinizdeki parçalardan buna uyacak bir başka parça yerine yerleşip eğer uygunsa ilk parça ile birleşir. Bu arada parçaları çeşitli yönlerde çevirerek inceleyebilirsiniz.

Bütün parçaları birleştirdiğinizde çok iyi bir şekilde korunmakta olan CENTRALIS reaktörüne giderek içeri girin. Burada REACTOR komutunu verdiğinizde reaktör odasına geçersiniz ve bütün ROD'ları yerlerine yerleştirirsiniz.

Oyun değerlendirilmesi:

GRAFİK	: 9
SES	: 6
ZORLUK	: 7
OYNANABİLİRLİK	: 8
GENEL	:

GEOS WRITER'S WORKSHOP

Bu ay sizlere Geos'la çalışan bir kelime işlemci programı, "Writer's Workshop" tanıtacağız.

Bu kelime işlemciyi kullanabilmek için her şeyden önce Geos disketine sahip olmanız gerektiğini hatırlatalım.

Geos boot disketini yükleyip kapattıktan sonra (disk menüsünde yeralan "close" şıkkını seçeceksiniz) Writer's Workshop disketini sürücüye yükleyip açmanız (aynı menüdeki "open" şıkkını seçmeniz) gerekecektir.

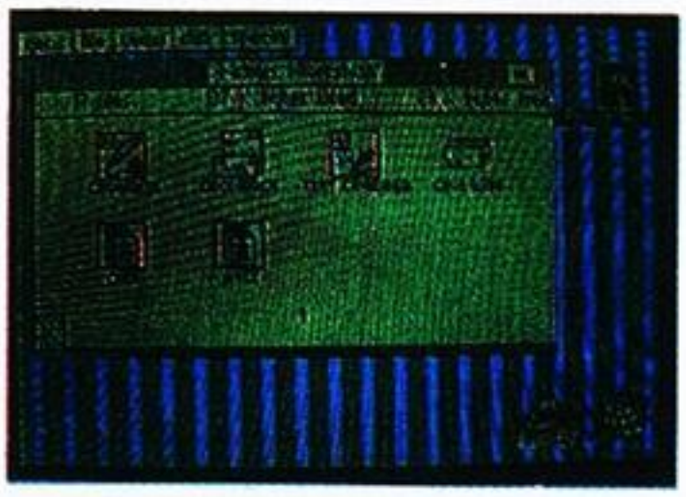
Writer's Workshop disketini açtığınızda karşınıza çıkacak ilk sayfada "geowrite", "geoMerge", "text grabber" ve "geolaser" programları yeralmaktadır. GeoWrite şıkkını seçmek için şeklin üzerinde joystick'i bir kez ateşleyip "file" menüsüne gitmeniz, oradan da "open" komutunu seçmeniz gerekecektir (veya şekil üzerinde joystick'i iki kez ardarda ateşleyin). GeoWrite yüklendikten sonra ilk ekranda yeni döküman yaratma (Create-new document); varolan dökümanı açma (open-existing document); ana çalışma ekranına çıkış (Quit-to deskTop) seçenekleri yeralacaktır. Yeni döküman yaratmak için "create" şıkkını seçtiğinizde karşınıza çıkacak dialog kutusu size yeni dosyanın ismini girmenizi isteyecektir. İsmi girip return tuşuna basın.

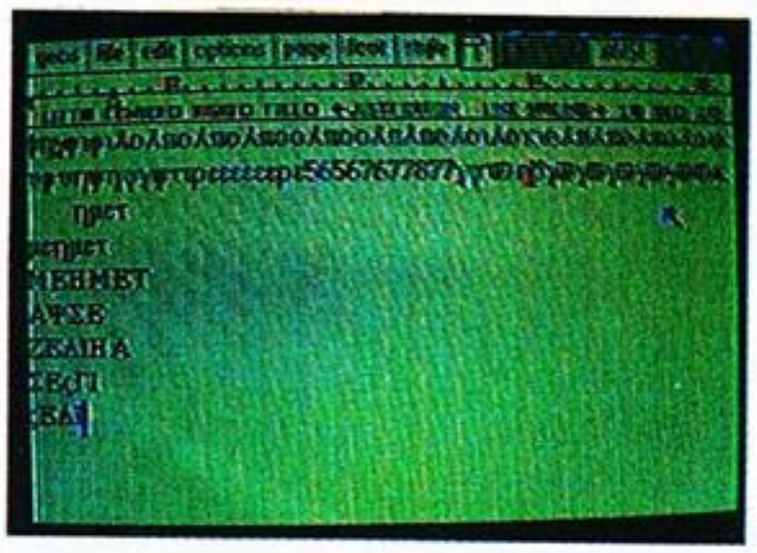
Varolan dosyayı istediğinizde çıkacak olan dialog kutusu, diskette yeralan GeoWrite dosyalarını dökecek; birini seçip "open" komutuyla açmanızı isteyecektir.

Döküman oluşturabilmek için ekranda görünen yazı imlecini yerleştirip klavyeden harf girmeniz yeterli olacaktır. Yazı alanı toplam olarak 6 inch genişliğindedir.

GeoWrite ekranındaki kumanda tablosundaki şu başlıklar altında toplanmıştır: Geos-file-edit-options-page-font-style-sayfa numarası, gösterge kutusu, dosya ismi.

File-dosya-menüsünde yer-





alan komutlarla bir dosyayı kapayabilir, dosya üzerinde yaptığınız değişiklikleri kaydedebilir; metni ekranda yerleştirebilir, yeni isim takabilir; dosyayı yazdırabilir, sayfanın ekrandaki görüntüsünü alabilirsiniz. Edit-yazma menüsündeki komutlarla (cut-copy-paste) yazınızdaki istediğiniz bölümleri kaydırabilir, aktarabilir, kopyalayabilirsiniz.

Kumanda çubuğundaki "options" şıkkı içinde yeralan "search" komutu sözcükleri veya kalıpları istediğiniz yeni sözcüklerle değiştirmenizi sağlayacaktır.

Cetveli kullanarak yazının sınırını ve satır aralığını belirleyebilir, cetvelin sol tarafında yeralan "M" ve "P" harfleriyle satırın ve paragrafın cetvel üzerindeki yerini düzenleyebilir, "Justification" komutunu kullanarak yazının kâğıt üzerindeki yerini ayarlayabilirsiniz.

Yazı karakterlerine gelince: 6 değişik karakter kullanabilir; karakterleri yedi değişik biçimde yazdırabilirsiniz.

"Options" menüsünde yeralan "open header" komutuyla yazıya başlık hazırlayabilir, işiniz bittikten sonra "close header" şıkkını kullanarak başlığı kapatabilirsiniz.

Dipnot yaratmak için "Options" menüsünde yeralan "open footer" komutunu kullanabilirsiniz. Sayfa numarasını, tarihi ve zamanı başlık içinde veya dipnotta gösterebilirsiniz.

GeoWrite'ın en önemli özelliklerinden biri, GeoPaint'le yarattığınız dökümanı "Paste" komutunu kullanarak yazdığınız metnin içinde kullanabilmeniz. Yerleştirme işini yazı imleciyle yaptıktan sonra "hide pictures" komutuyla resmin görünmesini ve dökümanınızda yer kaplamasını önleyebilirsiniz.

"File" menüsünde yeralan "print" komutuyla dökümanınızı bastırabilirsiniz. Yazıcı seçimini yapmak için asıl geos disketini kullanmanız gerekecektir.

Dökümanınızın birden fazla kopyasını çıkarmak istiyorsanız "writers workshop" disketinde yeralan "GeoMerge" programını kullanabilirsiniz. Çıkardığınız her bir kopyaya ayrı isim veya adres verebilirsiniz. GeoMerge kullanabilmek için önce bir döküman yaratmanız gerekecektir.

Diskette yeralan "Text Grabber" programıyla Commodore'la kullanılan belli başlı kelime-işlemci programlarla hazırlanan data dosyalarının geos'ta kullanılması ve geowrite data dosyası haline gelmesi mümkün oluyor. □

OSAM BİLGİSAYAR

SÜPER OYUNLAR SERİSİ

OSAM

SÜPER OYUNLAR PAKETİNDEN

9 PAKETİ BİRDEN ALANLARA

"LIGHT PEN"

HEDİYE EDİYOR

PAKET PROGRAMLARI.....5500 TL.
1 KASETTE 20 OYUN.....2200 TL.
1 KASETTE 40 OYUN.....3000 TL.
1 KASETTE 1 OYUN.....1500 TL.
COMMODORE T-SHIRT.....1500 TL.
LIGHT PEN.....10.000 TL.

1986 YILININ SÜPER PAKET PROGRAMLARININ ÜRETİMİ DEVAM ETMEKTEDİR.

10 SÜPER PAKETTEN BEŞ PAKETİ TOPTAN ALANA % 20 İNDİRİM UYGULANMAKTADIR.

Süper Paket Programlar serisine devam

PAKET 7

KAFA AYARI PRG.
EXPLODING FIST II
1942
ALLEYKAT
BOMBA
CYBORG
FISTFULOF BACK
HACKER II
MISSION A.D.
DRAGONS LAIR
TAİ BOXING
INFILTRATOR I
INFLITRATOR II
INFILTRATOR II
SATANAS HALLOW
JACK THE NIPPER
SPIKY-HAROD
SHOW JUMPING
SPINDIZZY
CORE
BOUNCES

PAKET 8

KAFA AYARI PRG.
YIE AR KUNG-FU II.
STRIKE COBRA
TARZAN
AVENGER
FAKLAN'S 82
BASILDON BOND
NEMESİS
WARHWK II.
PARKPATROL
BURGER TIME
CRACKER REVENGE I
CRACKER REVENGE II
WARIOK
PAPERBOY
FORCE
GREMLİNS
CASİNO ROYALE
GANDALF SORCERER
GALİLAN
STAR PİNG-PONG

PAKET 9

KAFA AYARI PRG.
FOOTBOL OF YEAR
STRET SURFER
SOLDIER
SABRE WULF
BUCK BORGERS
F-FOX
BREAKTHRU
DANTES INFERNO
GREAT PACMAN
CLUMSY COLINS
TWO ON TWO
BAZOOKA BILL
CAPTURED
ARAC
WILL O'THE WIPS
CUTHBERT TOMBS
ORANGE SQUASH
BOZOS NIGHT
HOODOO VOODOO
BREAK FEVER

PAKET 10

KAFA AYAR PRG.
HE-MAN
HE-MAN (ADVENTURE)
STALLONE COBRA
SHAD' LIN'S ROAD
MUTANTS
SOLDİER ONE
IT'S KNOCK OUT
THE YOUNG ONES
SUPER CROSS
MR DO'S CASTLE
FLASH BEEAR
RALLY SPEDWAY (2)
GHOST CHASER
MAD NURSE
FREAK FACTORY
ANTRIAD GAME
LEGENT OF KAGE
SOLDIER FORTUNE
TENTH FRAME
WILLIAM WOBLER

BAYİLİKLER VERİLECEKTİR

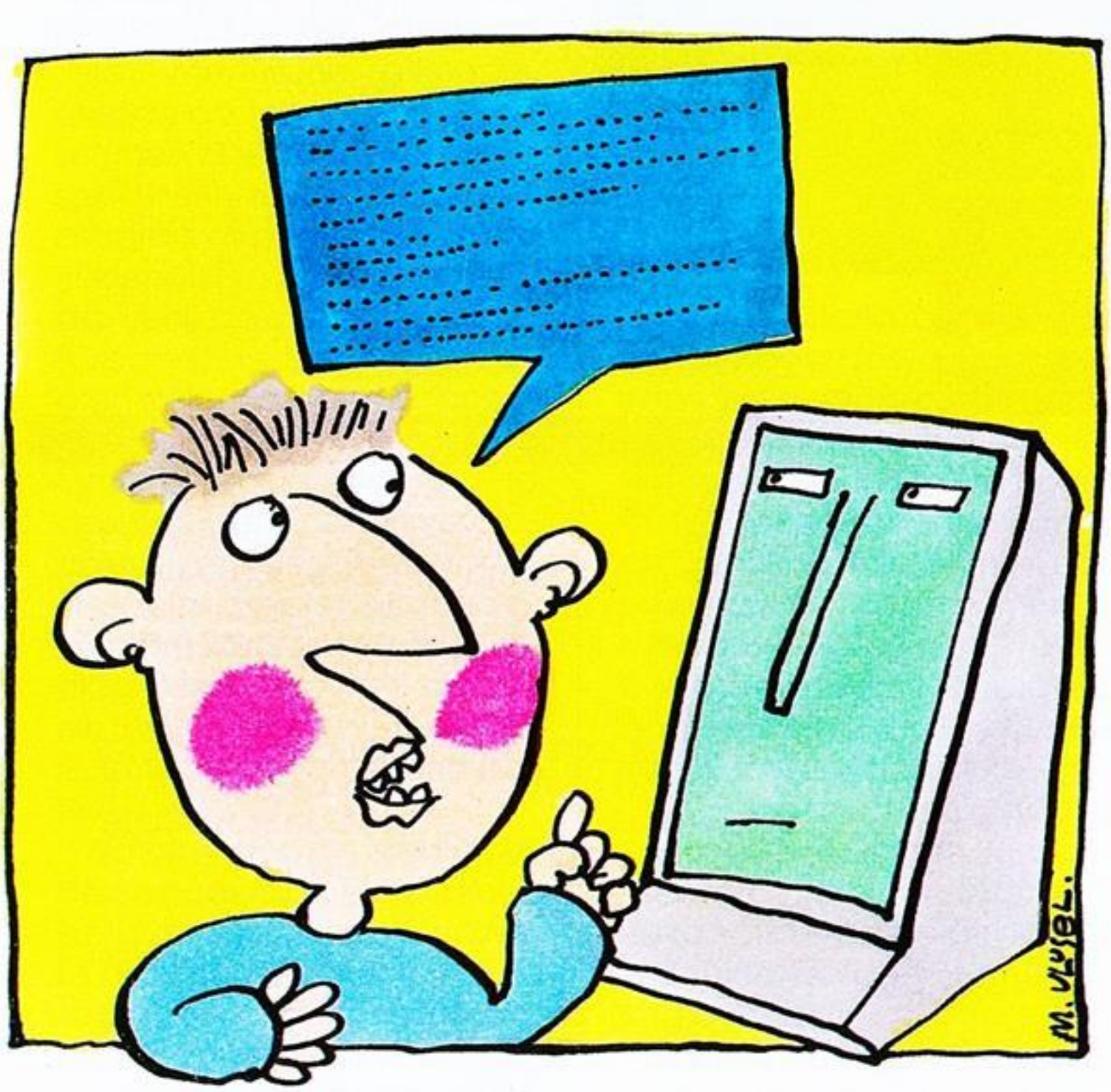
Ebuzziya Caddesi Sakızlı Yalı Sokak No.3 Bakırköy-İstanbul Tel: 583 72 46

Makine diline merhaba (5)

AYHAN KALAYLIOĞLU

4. MAKİNE DİLİ PROGRAMLARIN GİRİLMESİ

Yazı dizimizi buraya kadar takip etmiş olanlarınız için bir özetleme yapalım. Dizimiz şimdiye kadar üç ünite işledi. Bunların ilki "Giriş", ikincisi "6510 Ana İşlem Ünitesinin Tanıtımı", üçüncüsü '6510 Ana İşlem Ünitesinin Komutları ve Adresleme Tipleri" idi. Üçüncü bölümün sonunda artık ana işlem ünitemizi tanıyoruz ve bunun tüm komutlarını ve fonksiyonlarını gördük. Ee artık bir şeyler yapmanın zamanı geldi değil mi?.. Bu dördüncü bölümde sizlere, makine dili programlarının nasıl girilip yürütüleceğini göstereceğiz. Aslında, programı doğru yöntemle girip çalıştırmak, programı yazabilmek kadar önemli. Diyelim ki, kısa bir program parçası deneyeceksiniz; böyle durumlarda monitor programını kullanmak ideal bir çözüm. Tam aksi durumu gözden geçirelim. Diyelim ki, çok uzun ve ciddi bir program yazacaksınız ve hattâ bu programı profesyonel amaçlarınıza uygun olarak yazacaksınız. Böyle bir durumda monitor programı kullanıp, bu sorunun altından kalkmak, insanoğlu denilen varlığın sınırlarını aşan bir olay. Kısacası, daha işin başlarındayken ipin ucunu kaçırırsanız; ipin ucunu yakalamak için ısrar ederseniz bu sefer de aklınızı kaçırma tehlikesi baş gösterir. O yüzden bu bölümü çok dikkatli okuyun ve uygulamaya çalışın. Bu bölümde, size sistemli çalışabilmeniz için bazı öneriler getirmeye çalışacağız. Bunlar, fazla detaylı ve yolu uzatan metotlar olarak görünebilir. Siz kendinize uygun değişiklikleri yapmakta serbestsiniz. Fakat, sizden ricamız buradaki önerilerimizi kulak ardı etmemeniz.



Bir önceki bölümde gördüğümüz gibi, makine dili programlar, emir kodlarından ve bunlara ait operandlardan oluşmuş bir bilgi yığınıdır. Yapılan bir işlemi daha yakından takip etmek için, dilerseniz önce BASIC'te ufak bir örnek program yazalım. Bu programın yapacağı işlem, ekran hafızasına doğrudan bir şey yazmak ve renk hafızasında da bu karakterin rengini koymak olsun. Pek açık olarak görüldüğü gibi bu işlem için BASIC'teki POKE komutunu kullanacağız.

```
POKE 1024, 1: REM "A'nın ekran
POKE 55296,7: REM Sarı renk kodu
```

Bu komutları verdiğimiz anda, ekranın sol üst köşesinde sarı renkli bir 'A' harfi belirir. Şimdi, aynı işlemi makine dilinde nasıl yapacağımızı düşünelim. Hemen, hafızamızı çalıştırarak POKE komutunun makine dilindeki karşılığının 'STA' olduğunu hatırlarız. Hatırlayacağımız üzere STA komutu, akümülatörün içeriğini belirtilen adrese saklıyordu. Fakat, önce akümülatöre gerekli değeri yüklemek gerekiyordu. Yani:

```
LDA   1
STA 1024
```

Burada, aküye '1' değerini yükleyip, bunu da 1024 adresli hafıza biriminde sakladık. Aynı kalıplara sa-

dık kalarak, renk kodunu da oturtabiliriz:

```
LDA #7
STA 55296
```

Eğer, bu komutları doğrudan bu şekilde verirsek, sadece '?SYNTAX ERROR' elde ederiz. Bunun nedeni ise Commodore 64'ün direkt olarak sadece BASIC komutlarının anlamasıdır. Yani şu anda izlediğimiz yol yanlış olsa gerek. Hangi yolu izleyelim peki?.. Hatırlayacak olursak, makine dili programlar hafızada peşpeşe dizilmiş emir kodlarından ibaretti. Önce, her bir komutun emir kodunu belirleyelim. Bu işlem için, küçük kitapçığımızı kullanacağız. Buradan, LDA komutunun direkt adresleme tipi için $A9 kodunun kullanılacağını görürüz. Bunun hemen arkasından operand byte'ı gelir. Yani $01. STA komutunda ise absolut adresleme tipini kullanmışız. Küçük kitapçığımız kullanacağımız kodun $8D olacağını söylüyor. Bunun da arkasından, kendisine ait operand gelmek zorunda. Şimdi ise operand, bir hafıza elemanının adresi. Yani 16 bitlik bir değer. Bu değeri temsil etmek için, iki adet 8 bitlik değere ayırmamız gerekiyor. İlk önce düşük değerli byte, arkasından yüksek değerli byte takip etmek zorunda. Onaltı bitlik bir değeri iki adet sekiz bitlik değere ayırma işlemi hekzadesimal sayı sistemi kullanıldığında çok daha rahat yapılabiliyor. O yüzden ilk önce onaltı bitlik değerleri hekzadesimale çevirelim. 1024 sayısı hekzadesimal olarak $0400'a karşılık geliyor. 55296 sayısının karşılığı ise $D800. Programımızı bir kere daha, bu sefer hekzadesimal sayı sistemini kullanarak yazalım.

```
LDA #$01
STA $0400
LDA #$07
STA $D800
```

Onaltı bitlik değerlerimizden ilki olan $0400'ın düşük değerli byte'ının $00, yüksek değerli byte'ının $04 olduğunu görüyoruz. Yani STA $0400 satırı, byte karşılıkları kullanılarak yazıldığında $8D, $00, $04 byte'ları dizisi olduğunu çıkardık. Benzeri bir analizle, programın geri kalan kısmının, yani LDA # $07, STA $D800 kısmını $A9, $07, $8D, $00, $D8 byte'larıyla temsil edebiliyormuşuz. Tüm programımız şu şekil almıştır:

$A9, $01, $8D, $00, $04, $A9, $07, $8D, $00,. $D8

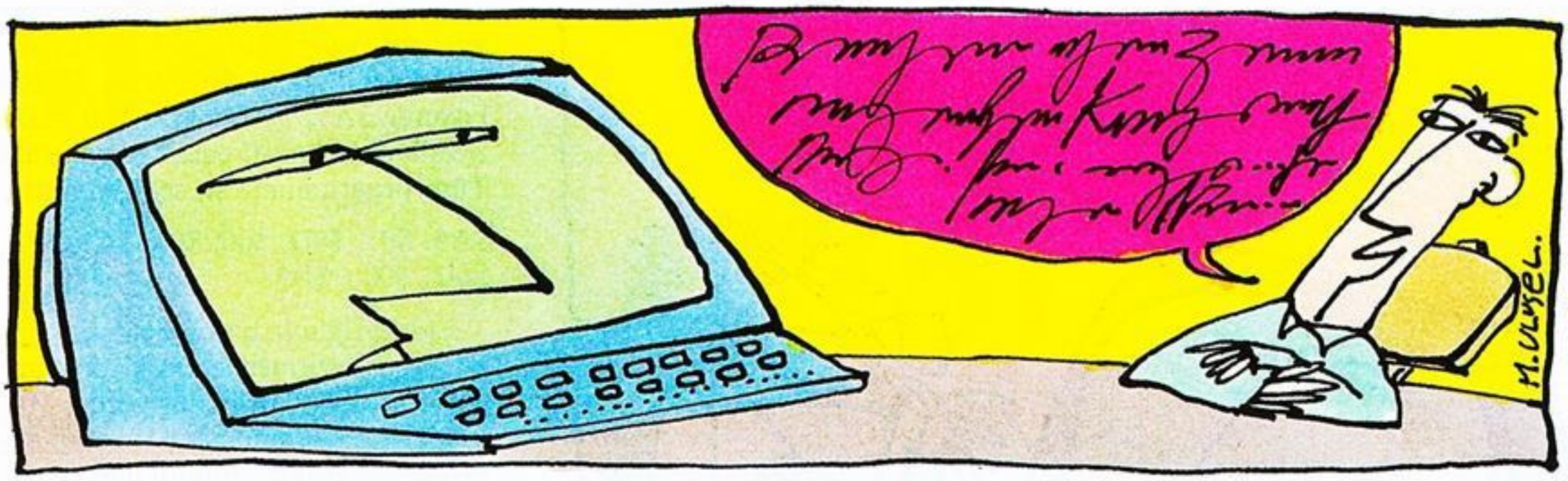
Elde ettiğimiz bu byte dizisini hafızaya yerleştirmemiz gerekiyor. Şimdi bir problemle karşılaşıyoruz. Makine dili programımızı hafızanın neresine koyacağız? Bunun için en uygun yer olarak, $C000'dan $CFFF'e kadar uzanan bölge göze çarpıyor. Bu bölgenin en önemli özelliği, işletim sistemi ve BASIC yorumlayıcısı tarafından hiçbir zaman kullanılmaması. Bu bölgenin genişliğinin 4 KByte olduğunu görüyoruz. Oldukça avantajlı bir bölge burası. Fakat, büyük ihtimalle, bu bölgede bir monitor programı çalıştırmak isteyeceksiniz ve monitor programı için ise en ideal bölge burası. O yüzden, ilk başlangıç programımız için daha değişik bir yer önereceğim. Burası, işletim sisteminin ve BASIC yorumlayıcısının erişebileceği bir bölge olmasına karşın, diğer programlarınızla çakışma ihtimali daha az. Ayrıca, bu bölgenin BASIC yorumlayıcısı tarafından bozulması ihtimali de çok uzak. Bunun için oldukça uzun bir BASIC programının oldukça çok işlem yapması gerekiyor ki bu bölgeye erişsin ve programınızı bozsun. Şu anda yazmakta olduğumuz ilk deneme programımızda herhalde böyle bir durum sözkonusu olmayacağı için gönlümüz rahat etsin! Şimdi, hekzadesimal modda, baştan aşağı yazdığımız programımızı desimale (ondalığa) çevirelim ki BASIC'ten bu kodları rahatça girebilelim. Çünkü BASIC yorumlayıcımız, sadece ondalık sayıları anlıyor ve bunlarla işlem yapıyor.

169,1,141,0,4,169,7,141,0,216

Şimdi ufak bir BASIC programının bu programı bizi için hafızaya yerleştirmesini isteyelim:

```
100 FOR I=0TO9
110 READ A:POKE 6*4096+I,A
120 NEXT
130 DATA 169,1,141,0,4,169,7,
                    141,0,216
```

Bu programı yazıp RUN komutuyla başlattığımızda, makine dilindeki programımız 6*4096 (=$6000=



24576) adresinden itibaren belleğe yerleşir. Eh artık makine dilindeki programımızı başlatabiliriz. Bunnun için BASIC'teki SYS komutunu kullanacağız. Şimdi SYS komutunu verip, başlangıç adresi olarak da 6*4096 verirsek makine dilindeki programımız, BASIC yorumlayıcısı tarafından bir alt program olarak çağıracak ve BASIC yorumlayıcısına geri dönmesi beklenecektir. Aman! Dikkat! Az kaldı bir şey unutuyorduk! Diyelim ki ana işlem ünitesi STA $D800 komutunu da yürüttü. Daha sonra ne yapacak? Valla, bu durumda belli değil. Çünkü, bizim yerleştirdiğimiz küçük makine dili programının arkasında neyin yeraldığını bilmiyoruz. Büyük ihtimalle, bu bölgede gelişi güzel içeriğe sahip düzensiz byte değerleri bulunacaktır ve bunların arasında, ana işlem ünitemizi 'crash' ettirecek komut kodları da şans esiri bulunabilir. Böyle bir durumda, bilgisayarı, kapatıp açmak veya RESET'lemekten başka çareniz yok. Tabiî ki bu olay, ilk makine dili programımızı yazarken pek hoş olmayan bir tecrübe.

Ana işlem ünitesi dört adet komutumuzu yürüttükten sonra, kontrolü BASIC yorumlayıcısına geçirmelidir. Çünkü, SYS komutu bizim makine dili programını bir altprogram olarak çağırmıştı. Altprogramın geri dönmesi için programımızın sonuna bir adet RTS (return from subroutine) komutu, yani $60 (=96) kodu eklememiz gerekiyor.

POKE 6*4096+10,96

Böylece, programımızın sonuna (10'uncu byte'ı olarak) RTS komutunu ekledik. Artık programımızı

SYS 6*4096

komutuyla başlatabiliriz!

Derhal, ekranın üst sol köşesinde sarı renkli 'A' harfi belirir ve bilgisayar tekrar 'READY.' mesajı vererek BASIC yorumlayıcısına kontrolu verir. Eğer, ekranın sol üst köşesindeki sarı 'A' harfini bir türlü çıkaramıyorsanız, komutlarınızı hep ekranın en alt satırından veriyorsunuz demektir. Yani sizin sarı 'A' sol üst köşede beliriyor, fakat bilgisayarın kendi mesajları (READY, gibi) yüzünden ekran birkaç satır yukarı kaydırılıyor. Böylece, siz kendi yarattığınız sarı 'A' harfini bir türlü göremiyorsunuz. Bu yüzden, komutlarınızı ekranın en alt satırından vermeyin ki ekran kaymasın.

Şimdi, aklınıza şöyle bir soru gelebilir: Şu ufacık işlemi yapmak için bu kadar uzun uğraşmak mı gerekiyor? Çünkü burada yapılan işlemlerin çoğu, kodların aranması, operandların belirlenmesi ve sayı sistemlerinin çevrimi. Aslında, görüldüğü gibi, bunların tümü rutin ve belirlenmiş işler. Ee peki, bu işlemleri bilgisayarın kendisi yapamaz mı? Yoksa, bilgisayarın görevi ne?

Yani, sonuç olarak öyle bir yardımcı programa ihtiyacımız var ki, 'LDA # 1' gibi bir makine dili komutunu alsın ve tüm ara işlemleri kendiliğinden yapıp, sonuçta meydana çıkacak kodları otomatik olarak hafızaya yazacak. Gerçekte, bu işi yapan programlar mevcut ve bunlara 'ASSEMBLER' adı veriliyor. MART 87 sayımızda TUNA ERTEMALP arkadaşımızın verdiği SUPER ASSEMBLER (nam-ı diğer HYPRA-ASS) programı bu sınıf programlara kusursuz bir örnek oluşturuyor. Daha ilk baştan, assembler kullanmak yerine, amacımıza adım adım yol almayı daha uygun buluyorum. O yüzden önce monitor programından başlayalım.

'Monitor' kelimesi, İngilizcede 'izlemek', fiilinin karşılığıdır. Bu programın yaptığı işlem ise, bilgisayarın ana işlem ünitesinin registerlerini ve belleği yürüyen programla paralel olarak izlememize olanak veriyor. Tabiî, bu arada müdahale ederek gerek ana işlem ünitesinin registerlerini gerekse hafıza birimlerinin içeriklerini değiştirmek mümkün. Makine dili programları, monitor programından başlatmak da olası. Ayrıca, hafızanın bilumum acayip yerlerinde yeralan verilerinizi veya makine dili programlarınızı saklamak veya yüklemek istediğinizde, bu işi en rahat yapabileceğiniz yardımcı programdır monitor. Diyelim ki makine dilindeki programınız tüm hatlarıyla meydana çıktı, (içindeki adresler ve referanslar) böyle bir durumda programınızı monitorun line assembler'ı yardımıyla girebilirsiniz. Onaltılık sistemdeki verileri de belleğe doğrudan yazabilirsiniz. Bu son kullanılan yöntemle birçok programın üzerindeki ufak değişiklikleri yapmanız oldukça kolay ve kısa zamanda gerçekleşen bir olay haline geliyor. Diassembler denilen kullanım sayesinde hafızada yeralan bir makine dili programının listesini almak mümkün. Yazıcınız varsa, bu listing'i kâğıda da dökebilirsiniz. OCAK 87 sayımızda, sözünü ettiğim işlemleri yapan bir monitor programı, kullanımıyla birlikte vermiştim. Yakın bir zamanda, SMON adlı başka bir monitor programı daha vereceğim. Bunlar, yardımıyla birçok programı inceleyebilir, değiştirebilir ve yürütebilirsiniz. Gerçekten de, makine dilini en iyi öğrenme, başkalarının programlarını incelemekle gerçekleşir.

MPS 1000

Bütün bayilerde

Bilgisayarlı Eğitimden [redacted] NE HABER? [redacted]

EDWARD FISKE *Çeviren: Ayşenur Doksat*

Ülkedeki birçok halk okulu gibi Massachusetts, Provincetown Emekli Subay Çocukları İlkokulu da parasal açıdan sıkışık durumda. Buna rağmen ekonomi zihniyetli yönetim kurulu, okul müdürü Eliot Tocci'ye kısa bir süre önce şaşırtıcı bir şey sordu: "Gelecek yılki bütçenizde bilgisayarlara ayrılmış yeterli paranız var mı?" Bu önemsiz olay bilgisayarların ilk ve orta okullardaki rolünün dramatik değişimini gösteriyor.

Bilgisayarların büyük ölçüde kendileri birer bilgisayar delisi olan öğretmenlerin gayretleri sayesinde dershanelerdeki yerini sağlamlaştırmasından yaklaşık on yıl sonra -özellikle de son bir yıl içinde- asıl yerlerini bulmakta. Hardware (donanım) ve Software (yazılım) satışları giderek artıyor; ülkedeki birçok okul "bilgisayar cahilliği"ne karşı mücadele veriyor. Öğrenciler makinenin başında geçirebilecekleri daha çok zaman istiyorlar; öğretmenler bir haftalarını bilgisayar başında geçirdikten sonra kendilerine daha çok becerikli hale gelen üçüncü ve dördüncü sınıflara üstünlüklerini koruyabilmek için uğraşıyorlar.

Bazı kuruluşlar (PTas) bilgisayar alımını kamçılayabilmek için çeşitli araştırmalar yaparak yeni yeni pazarlar yaratıyorlar. Örneğin, Philadelphialı ünlü cazcı Grover Washington J.R., çocuklarının okuluna bir bilgisayar alabilmek için kendi orkestrası ile plak yapma fırsatını açık arttırma ile sattı.

DEĞİŞEN TAVIRLAR

Bir yandan da, bu alanda bariz bir tavır değişikliği gözlenmekte. Ebeveynler, öğretmenler ve yöneticiler, bilgisayarların yalnızca ek bir iş imkânı anlamına gelmediği Amerika'da toplumsal hayatın her alanının artık bilgisayarsız tasarlanamaz bir şey haline geldiği gerçeğine uydular. Eğer okullar gerçekten çocukları hayata hazırlıyorlarsa öğrencilerin bu yeni teknolojiyi rahatça kullanabildiklerini de göstermek zorundalar. Birleşik Devletler Eğitim Departmanı üyesi Linda Roberts şöyle diyor: "Herkes çocukların bilgisayarı tanımak zorunda olduğuna inanmak durumda."

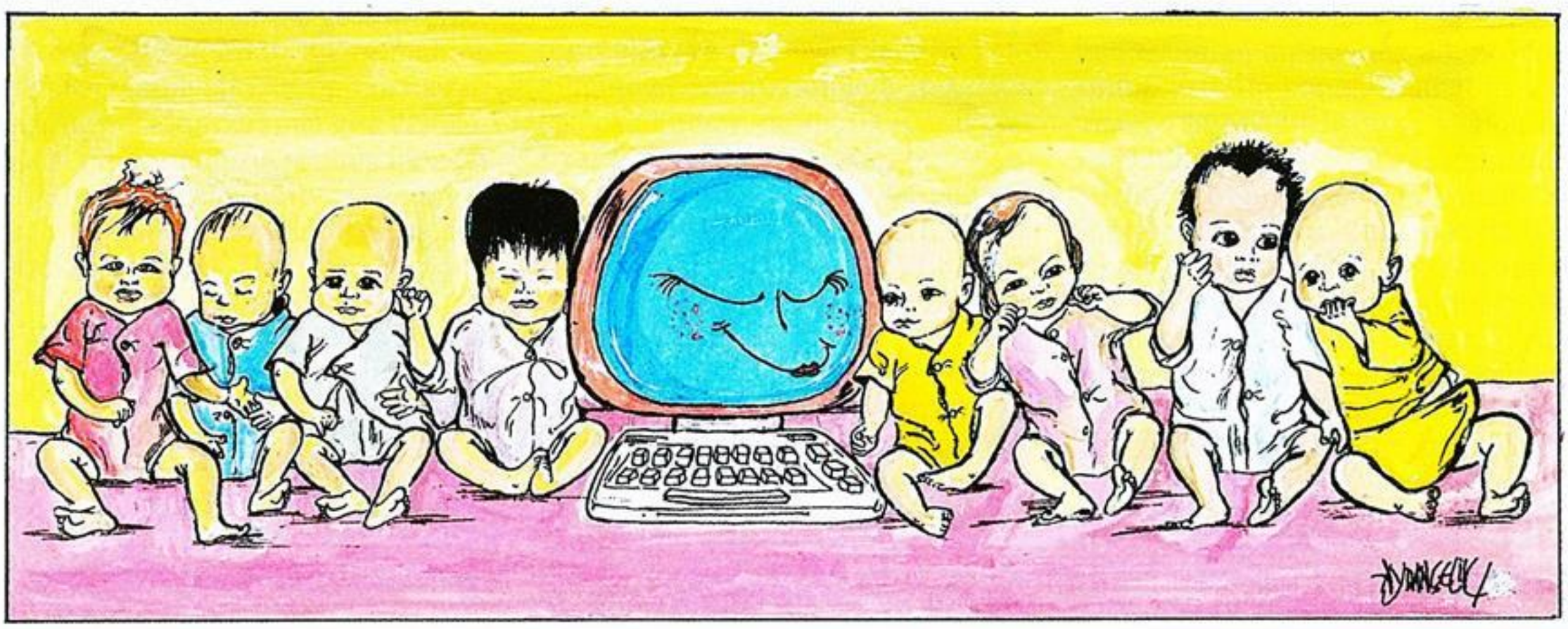
Washington Milli Eğitim Müdürü Frank B.Brouillet ise şöyle diyor: "Bilgi-işlem devrimi başladı. Ne kadın ne erkek, hiç kimse bunu durduramaz. Çocuklarımız bu devrimin ya galipleri ya da mağlupları olacaklar."

Bilgisayar karşısındaki bu yeni tavrın Amerikan okulları üzerindeki etkisi istatistiklerde görülebilir. Bir araştırma şirketine göre 1982 yılında 77.000 okuldan yaklaşık 23.000'inde mikrobilgisayar vardı. Bu, liselerin %50'si ortaokulların %40'ı ve ilkokulların %20'si demektir.

Asıl çarpıcı olanı ise bu rakamların büyüme hızı. 1981-1982 yıllarında bu teknolojiye sahip okulların sayısı liseler için %33, ortaokullar için de %48 oranında arttı. Bilgisayarlı ilkokulların sayısı ise yaklaşık iki katına çıktı. Araştırmalar 1982 yılında mikrobilgisayarı olan okullardaki öğrencilerin makine başında 9 saatten fazla zaman geçirdiklerini göstermektedir.

Okullarda bilgisayar kullanımını kanuna bağlayan ilk eyalet, 1981 yılında Florida oldu. 229.8041 sayılı yasaya göre "öğretme ve öğrenmeyi daha etkili ve verimli hale getirmek ve eğitim programlarını topluma daha uyumlu kılmak için bilgisayar teknolojisini kullanmak eyalet politikasıdır."

Okulların bilgisayara yaklaşımı hizmet verdikleri topluluklar kadar çeşitlilik gösteriyor. Cope Cod'daki Emekli Subay Çocukları İlkokulu küçük yatırımlar yapan bir okul olarak



iyi bir örnek. Okul geçen yıl bir bilgisayar aldı, ancak kullanmayı bilen kimse yoktu ve Elliot Tocci, üç öğretmeni Boston'daki bir haftalık bilgisayar kursuna gönderdi. Bu öğretmenler de diğer 18 kişilik öğretmen kadrosuna bu makineyi kullanmasını öğrettiler. Diğer öğretmenler 6'şar kişilik gruplara bölündüler; 3 öğretmenden her biri, her bir grubun ilk öğretimi için yarım saatlik bir süre kullandı ve ardından gruplarını değiştirerek denetimli pratikler ve ödevler yaptırdılar.

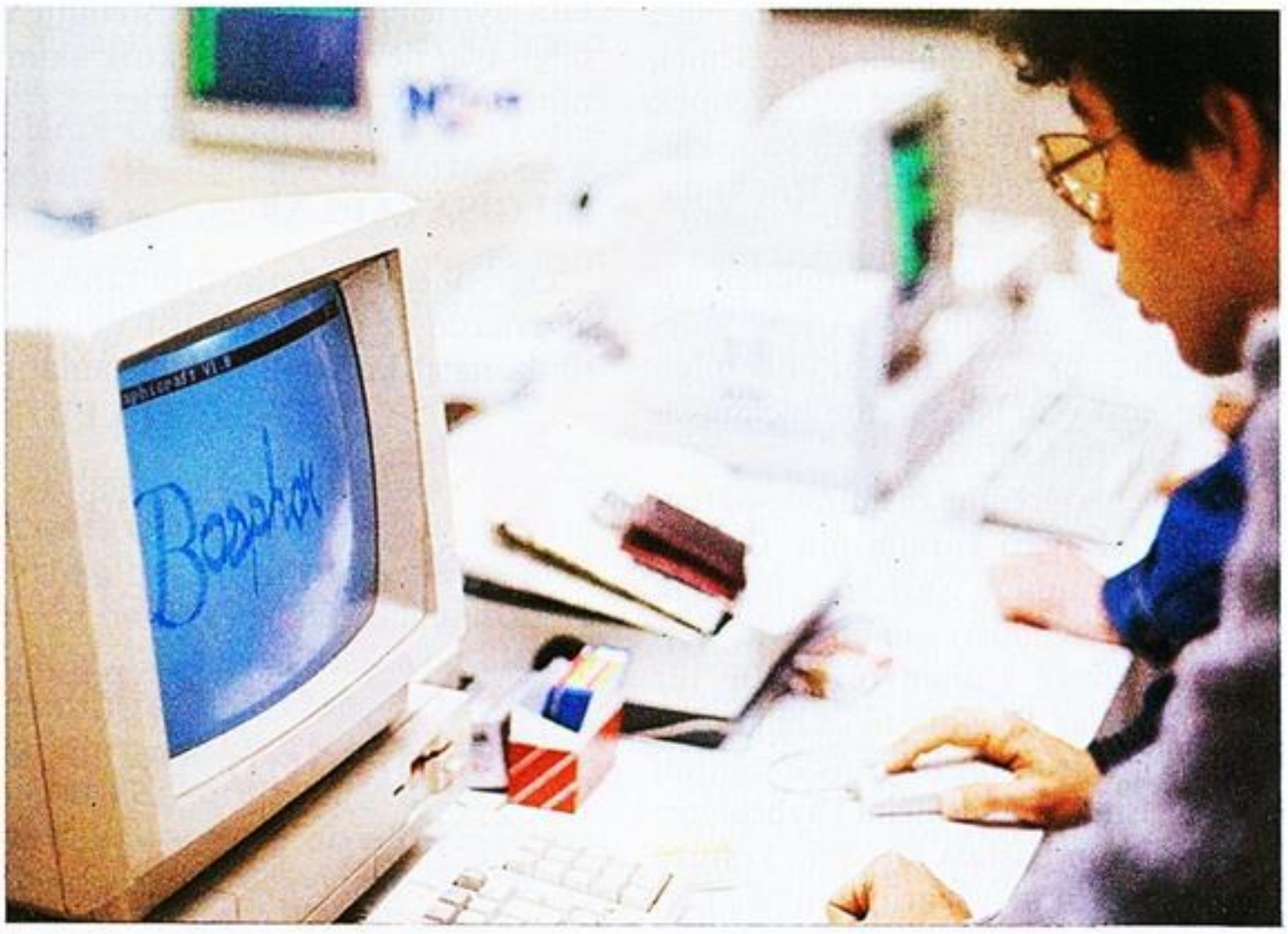
Daha sonra bilgisayarın kendisi dershaneden dershaneye taşındı ve geçen yılın sonunda 10 sınıfın her birinde ikişer hafta süreyle kalmış oldu. Anaokulundaki çocuklar dahil, herkes bazı pratik bilgileri öğrendiler ve çok geçmeden bütün üçüncü sınıf öğrencileri kendi programlarını yazar hale geldiler. Çocuklar ihtiyaç duydukları durumlarda okuldan önce ya da sonra kütüphaneye gidebiliyorlardı.

Müdür Tocci tek bilgisayarın bile daha heyecanlı bir öğrenme ortamı yarattığını yalnızca "soyut kavramları somutlaştırmaya" yardımcı olmadığını aynı zamanda "insanı yaratıcı düşünme biçimleri araştırmaya yönelten beklenmedik sonuçlar" ürettiğini ve böylece "sınıfa bir tazelik, heyecan ve özgürlük" getirdiğini söylüyor.

Spektrumun diğer ucunda ise ülkedeki en verimli ve yenilikçi bilgisayar eğitim programının ne olabileceğinin peşinde koşan Houston Bağımsız Okul Bölgesi bulunuyor. Houston okullarının baş müdürü Billy R.Reagon, bilgisayar dilinin okuma, yazma ve aritmetikle birlikte temel disiplinlerin katıldığına inanıyor. "Öncelikle anaokulu öğrencilerini eğitmekle başlıyoruz. Bilgisayar dilinin ilk beş yıl içinde yalnızca okumaya yardımcı olacağına inanıyoruz" diyor.

Houston bölgesi, bilgisayarı çeşitli amaçlar doğrultusunda kullanıyor. Öğrenciler bilgisayar destekli programlarla terminallerde yazarak okumayı öğreniyorlar. İngilizce konuşmayan küçük bir grup öğrenci İngilizce derslerinde konuşan bilgisayarlarla çalışıyorlar. Bölge mütevelli heyeti bilgisayar dilini öğrenmenin bir mezuniyet şartı olmasını onayladı.

New York'ta bir okulda Einstein'in rölativite teorisini öğreten bir program geliştirildi. Adı Einstein olan



programa öğrenciler ağırlıklarını söylemekle işe başlıyorlar; bilgisayar "MHP (Mile Per Hour-Saatte alınan mil) olarak hızınız nedir?" diye soruyor. Öğrenciler daha büyük hızlara çıkarak kütlelerinin ne kadar arttığını anlayabiliyorlar. Örneğin 165 poundluk bir öğrenci 100.000 MPH hızla hareket ederse bir bilgisayar şöyle karşılık veriyor: "Üzgünüm. Kazandınız. Şimdi 165.00002 poundsunuz. Eğer daha hızlı hareket edecek olursanız ağırlığınız ciddi bir sorun haline gelecek. Daha hızlı hareket etmek ister misiniz?" Aynı öğrencinin ağırlığı ışık hızından 1 MPH daha az bir hızda 4,087,096,15 pounda yükseldi. Eğer öğrenci daha hızlı hareket etmeye niyetlenmezse bilgisayar öğrencisinin kullandığı yakıt cinsini sorar, bunu kilowatt-saate çevirir ve öğrenciye bu hipotetik yolculukta kullanılan enerjinin "Bir yıl süreyle New York City'ye koşmak için yeterli olan enerjiden daha fazla olduğunu" belirtirdi.

Princeton İlkokulu'nda bilgisayar müzik derslerinde sesler çıkarıyor. Do majör anahtarını belirledikten sonra çeşitli tonlardaki gamların "haritası"nı öğrencilere yaptırıyor, kullanılacak nota sayısı temelinde ritmleri çıkarttırıyor ve programları yazdırıyor. Böylece bilgisayar "Daha dün annemizin kollarında yaşarken" şarkısını ve başka şarkıları çalabilecek şekilde programlanmış oluyor.

Bilgisayarların Amerikan okulları üzerindeki en dramatik etkisi özürlü öğrencilere yeni teknoloji dünyasının kapılarını açan özel bir eğitim alanı oldu. Neredeyse her sakatlık için bir devre anahtarı alet ya da bir adaptör mevcut. Bir öğrenci tekerlekli iskemlesinin aralığında küçük küp biçiminde bir şalteri kullanıyor. Bazıları da bir el ya da ayak parmağı hareketiyle veya bir göz hareketiyle bilgisayarlarına komut verebiliyorlar.

Bu bilgisayarların en iyi yanı özürlü öğrencileri öğrenmeye, araştırmaya teşvik etmesi. Çünkü tekerlekli iskemleye bağımlı olanlar, diğerlerinin kendileriyle ilgilendiklerine güvenmek zorundalar. Yaşamlarında özgür olarak kontrol edebilecekleri çok az şeyi olan bu öğrenciler bilgisayarları kendileri kontrol edebilirler.

BİLGİSAYAR DESTEKLİ EĞİTİM Mİ? BİLGİSAYAR EĞİTİMİ Mİ?

Amerikan okullarında bilgisayarların etkisini tanımlamak, neredeyse kullanıldıkları yerleri kategorize etmek kadar karmaşık bir iş. Bundan kısa bir süre önce bilgisayar destekli eğitimin bilgisayar dillerini öğrenmeyi engellediğini iddia ediyordu.

Böyle bir iddia hiç de şaşırtıcı değil; çünkü herhangi bir yeni teknolojik gelişim karşısında gösterilen ilk tepki yeni tekniğin halen ne yapılmakta ise yine onun yapılması için kullanılması oluyor. İlk sinemacılar kameralarını setlerin önüne yerleştirdiler ve oyuncuları aynı bir tiyatro oyunundaymışçasına setin bir yanından sokup çıkarttılar. Ancak sonraları sinemanın kendine özgü imkânlar sunduğunu anladılar. Benzer şe-

kilde bilgisayarlar da sınıflarda önce çarpım tabloları kelimeleri hecelemek ya da devlet başkentlerini öğretmek gibi, öğretmenlerin sürekli yaptıkları rutin işleri tekrarlamak için kullanıldılar.

Bilgisayar destekli eğitiminden hâlâ aynı şey anlaşılıyor. Ancak giderek daha çok sayıdaki eğitimci bilgisayarların yeni bir öğretim biçimi yarattığını farkediyor.

Bu yeni öğrenme biçiminde öğrenciler bilgisayarı kafalarının tıka basa doldurulduğu şeyleri yansıtmak için değil, kendileri için bir şeyler keşfetmek üzere kullanıyorlar. Bu tür uygulamalar yaygınlaştıkça bilgisayar destekli eğitim mi bilgisayar eğitimi mi tartışması güncelliğini kaybediyor.

Bugünün başlıca tartışma konusu bilgisayar okur-yazarlığının ne anlama geldiği. Bugünün teknoloji dünyası için gereken temel becerilerden birinin bilgisayarlarla daha rahat yaşayabilmek olacağı hakkında genel bir görüş birliği var.

Geleceğin Toplumu adlı kitabın yazarı Donald Michael, "Bilgisayar cahilliğinin insanları fonksiyonel olarak okuma-yazma ve aritmetik cahilleri kadar cahilleştireceğini" iddia ediyor.

Bazıları bilgisayar dilini öğrencilerin makineyi kullanma becerileri gibi bir şey olarak görüyor.

Diğerleri ise, kendi başına pratik becerinin dar ufuklu bir amaç olduğunu, yarının vatandaşlarının toplumsal ve teknolojik tezahürleri anlamadıkları sürece hastalıklı bir biçimde biraraya geleceklerini öne sürüyor. Bu görüşe göre "Okullara bilgisayar sokmanın ve çocuklara bu programları kullanmayı öğretmenin daha açık ve kolay bir yolunu bulmalıyız. Onlara bilgisayar fonksiyonlarının ve kullanma alanlarının neler olduğunu göstermeliyiz ki bundan sonraki yaşamlarında akıllıca seçimlerde bulunabilsinler."

Okul tipi bilgisayar eksperi ve Bilgisayar Dili üzerindeki Projektör adlı kitabın yazarı olan Ellen Richman yedinci sınıf derslerinin üç temel alanı içerdiğini söylüyor.

BİLGİSAYARLAR NASIL ÇALIŞIR?

Öğrenciler, kartları, devreleri ve diğer elektronik aksanı incelerler, parçalarını tanımlayabilmek için el hesap makinelerini digital saatlerini ve bilgisayar oyunlarını sökerek parçalara ayırırlar. İkili sayı sistemini ve bilgisayar devresinden elektrik akımının nasıl geçtiğini öğrenirler.

YAŞAMIMIZDAKİ BİLGİSAYARLAR

Öğretmenler bilgisayarın tarihçesini evlerde ve çeşitli endüstri alanlarında nasıl kullanıldıklarını anlatırlar. Sınıfta gruplar oluşturulur ve bu gruplar kendi aralarında bilgisayar kullanımı üzerine tartışmalara girerler; bu tartışmalar bilgisayar suçu ve gizliliğinden kariyerlere kadar bir dizi başlık altına sürdürülür.

Bilgisayar, deney yapma cesaretini artırdığı, çocuğun korkacağı riskleri ortadan kaldırdığı ve en önemlisi, son derece sabırlı olduğu için eğitimde vazgeçilmez hale geldi.

Bütün öğrenciler kendi tempolarında programlamayı öğrenirler. Richman, "Öğrencilerin kendilerinde bir ustalık hissetmelerini istiyoruz. Bilgisayarın kullanılabilecek bir alet olduğunu bilmelerini istiyoruz." diyor.

Geniş ölçüde bilgisayar kullanan ve alıştırma çalışmalarının ötesine geçmek için çaba harcayan okullar bilgisayarların eğitim biçimini önemli ölçüde değiştirebileceğini görüyorlar.

Bilgisayar kullanmanın eğlenceli olduğu açıktır. Öğrenciler bilgisayar kullanabilmek için ders saatleri dışında teneffüslerde bile uzun kuyruklar oluşturuyorlar. Örneğin, 10 yaşındaki Brett Wood "resim çizebildiği" gerekçesiyle bilgisayarı video oyunlarına tercih ediyor.

Bilgisayarların geleneksel okullarda çok ender rastlanan bir şey olan deneyi cesaretlendirici bir etkisi var. Uzmanlar bunun nedenini bilgisayarın eleştirel olmamasına bağlıyor. Bilgisayarların hata yapma riskini giderdiğini ve çocuğu "başka türlü almaktan korkacağı riskleri almakta özgür kıldığını" öne sürüyorlar.

Okullardaki bilgisayarlar öğrencilerin büyük ölçüde kendi tempolarında çalışmalarına izin veriyorlar. Bunun da sınıf ve düzey farklılıklarını giderici bir yanı var. "Bilgisayarlarla herbir öğrenciye kendi öğretim düzeylerini gösterebilirsiniz. Onbirinci sınıf öğrencileriyle kendilerinden daha hızlı ilerleyen üçüncü, sınıf öğrencileri birarada aynı sınıfta bulunabilirler." diyor bir öğretmen "Bundan sonra ne vasat düzeyde öğretim yapmak ne de zeki çocukların sıkılmamasını ya da yavaş çocukların geride kalmamasını dilemek zorundayız" diyor ve ekliyor: "Artık herkesin kendi temposunda ilerlemesine izin veriyoruz."

Özel eğitim öğretmenleri bilgisayarın yavaş öğrenenlere karşı son derece sabırlı olmak gibi bir avantajı olduğunu söylüyor.

BİLGİSAYAR ÖĞRETMENLERE MEYDAN OKUYOR

Bilgisayar, öğretmen-öğrenci ilişkilerini önemli ölçüde değiştiriyor. Çocukların çoğu bu bilgisayarlar hakkında öğretmenlerinin bilebileceğinden çok daha fazla şey biliyor. Gerçi bu dezavantaj; ancak öğretimin öğretmenleri zorlaması ile de fena bir şey olmayabilir.

Ancak eğitimciler bu tür zorlamaların bir bilgi edinme kaynağı olmaktan çok eğitim sürecini kolaylaştırıcı bir etkiye sahip olması gerektiğinde hemfikirler. Öğretmenlerin bilgisayarların sahip olmadıkları bir zeka deneyim ve muhakemeye sahip olduklarını hatırlamaları gerekiyor.

Bilgisayarlar onların asla olamayacağı kadar hızlı ve iyiler fakat onların gelecek gelişmelerinin rehberi hâlâ onlar. Önemli olan bir şeyi bilmedikleri zaman bunu çocuklara söyleyebilecek kadar akıllı davranmak.

Bazı eleştiriler de bilgisayar eğitiminin keskin bir dikkat gerektirmesinden dolayı okulun sosyal yaşamın-

da sorun yaratabileceğinden duyulan korkuyu açığa vuruyor. Buna karşın eğitimciler, pratikte bunun tam tersinin gerçekleştiğini öne sürüyorlar. Zamanlarının çoğunu mikrobilgisayarlarına yapışıp kalmış bir halde geçiren bir sürü "bilgisayar delisi" var; ancak bu tehlike hemen hemen her insan etkinliği için geçerli olan bir tehlike. Bilgisayar öncelikle öğrencilerin birbirlerinin ne yaptıkları ile ilgilenmelerini sağlıyor; bu nedenle onların da sosyal ilişkilerinin artmış olduğunu görüyoruz. Birbirlerinin omuzları üzerinden kaçamak bakışlarla bilgisayar odalarının etrafında dolanıyorlar program düzenlemelerinde birlikte çalışıyorlar ve bu programları paylaşabilmek için şebekeler kuruyorlar.

Bazı eğitimciler de bilgisayarların

Chagrin Falls'dan Ellen Richman da aynı fikirde; "öğrencileri pratik olarak mantıksal düşünmeye zorluyoruz. Eskiden çocuklara "söyleneni yapmıyorsunuz" derdik, şimdi "programı siz yazıyorsunuz-başlatıyor, planlıyor düzenliyorsunuz" diyoruz." diyor.

TOPLUM DESTEĞİ

Bilgisayar kullanımında deneyimli eğitimciler bilgisayar dili programlarının başarısının aileler içinde bulduğu desteğe bağlı olduğunu söylüyorlar. Ama hayli tutucu bir toplumda yaşıyoruz. Oysa ebeveynler çocuklarının neyle karşı karşıya olduklarını bir kez görürlerse yeni teknolojiye daha yardımcı olabilirler. Dahası bilgisayarları ebevynler için de bir yeniden-öğretim biçimi olabilir.

er ya da geç öğrencilerin düşünme ve öğrenme biçimlerini değiştireceğini öne sürüyor. Provincetown'daki yönetici öğretmenlerden biri Kristen Eastman bilgisayarın öğrencileri "çok özgün olmaya ve hızlı düşünmeyi geliştirmeye zorladığını" öne sürüyor. Örneğin kare yapmanın 8 aşamayı içerdiğini bilmeleri gerekiyor. Bu sürecin merkezinde çocuğun makine ile olan ilişkisinin bulunduğunu ekliyor. Eastman bilgisayarın aptal bir şey olduğunu ve kendi kendine hiçbir şey yapamadığını söyleyen bir çocuğu hatırladı: "Bir şeyler alabilmek için bir şeyler vermesi gerektiğini kavramak gerekiyor. Bilgisayar çocukta daha çok şüphecilik yaratıyor. Bu aktif bir katılımdır."

Houston okul bölgesi ebeveynlerin bilgisayar ödünç alarak çocuklarının özel öğretmenleri olabilecekleri bir sistem kurarak bir basamak daha ileri gitti. Örneğin geçtiğimiz Eylül ayında 32 yaşındaki bakım işçisi Eliseo Almaguer ve 6 yaşındaki kızı Evonne ve 12 yaşındaki oğlu Elises Jr. Texas Instruments'dan synthesizer'li bir bilgisayar alarak bunu modern teknolojiye ait tek işaretin eski bir TV olduğu iki yatak odalı mütevazi evlerine koydular. "Bazen bütün bir geceyi bu aletle geçiriyoruz. Çok eğlenceli oluyor." diyor Almaguer.

TOPLUMSAL SORUNLAR

Bilgisayarların okullarda hızla yaygınlaşması toplumsal ve etnik açıdan potansiyel olarak karmaşık sorunlar yaratıyor. Princeton'dan George Miller "en önemli sorunun bilgisayarın herkes için uygun bir araç olup olmayacağı" üzerinde duruyor.

Bu sorunun aciliyeti bilgisayar dilinin iş alanında uygulanabilirliği konusundan kaynaklanıyor. Sosyal bilimciler bilgi erişimin iktidar ve zafer için kritik bir öneme haiz olacağı bir "bilgi toplumu" üzerinde çalışmaya başladılar. Bu alanda çalışan bir araştırma ekibi pek yakın bir gelecekte işgücünün %75'inin işlerini yapabilecek bazı özel bilgisayar becerilerine sahip olması gerekeceğini iddia ediyor. Mahrumiyet bölgelerindeki okullar için bu becerilerin gerekli olduğu kuşku götürmez bir gerçektir. Çocuklara bu yeni teknolojiyi öğretmediğimiz takdirde yeni bir uyumsuz insan sınıfı yaratma tehlikesiyle karşı karşıyayız.

Okullar yeni teknolojiye giriş için ideal araçlardır. Ancak fakir bölgelerin bilgisayar donanımı alabilecek ve öğretmenleri eğitebilecek kadar yatırımın yapılıp yapılamayacağı da önemli bir sorundur.

Toplumsal önemi olan bir konu da bilgisayarın kız ve erkek öğrencilerin matematik fen ve teknik derslere yatkınlıklarındaki farkı ortadan kaldırma potansiyeli. Son on yılda yapılan araştırmalar kız öğrencilerin matematik ve pozitif bilimlerden korktuklarını -matematik endişesi- matematik derslerinde ve matematik gerektiren fizik derslerinde ileri çalışmalar yapmayı istemediklerini gösteriyor. Bazı lise bilgisayar klüplerinin erkeklerden yana olmasına rağmen bilgisayarların genelde bütün öğrenciler üzerinde olumlu bir etkiye sahip olduğu düşünülüyor. Sonuç olarak belki şu hikayeyi anlatmak gerekiyor:

Son bilim fuarında onuncu sınıf öğrencisi Victor Wong evindeki bilgisayarla geliştirdiği programı gayet yerinde olarak Bilgisayarla Öğretmenin Yer Değiştirmesi şeklinde isimlendirdi.

"Bir bilgisayar bir öğretmenden daha hızlı ve onun yapamayacağı ne yapabilir?" Bilgisayarın mavi-beyaz monitörü cevapladı: "O bir insandan daha hızlı düşünür. Hata yapmaz. Bir konudan fazlasını öğretebilir. Ve en önemlisi çıldırmaz." □

POPULAR COMPUTING, AĞUSTOS 1983'den

Commodore

BİLGİSAYAR NASIL ÇALIŞIR?

AYHAN KALAYLIOĞLU

Kesinlikle çoğu kimseler için bir merak konusudur; bilgisayarın kendi içinde neler karıştırdığı... Bu yazımızı dikkatlice takip edin ve olayları hayalinizde canlandırmaya çalışın. Göreceksiniz ki karmaşık bir muamma gibi görünen tüm bu olaylar iyi bir zamanlamayla uygulanan bir dizi mantıksal işlemlerden ibaret...

Bilgisayarlar artık hayatımıza iyice girmiş durumda. Büyük bir çoğunluk artık bilgisayarların ne yapıp ne yapamayacağını biliyor ve nerede nasıl bu aletlerden yararlanabileceğine karar verebiliyor. Bazı fanatik filmlerin de etkisiyle korkunç bir çehreye büründürülen bu aletleri artık hemen herkes tanıyor ve yalnızca çok iyi bir yardımcı olarak görüyor.

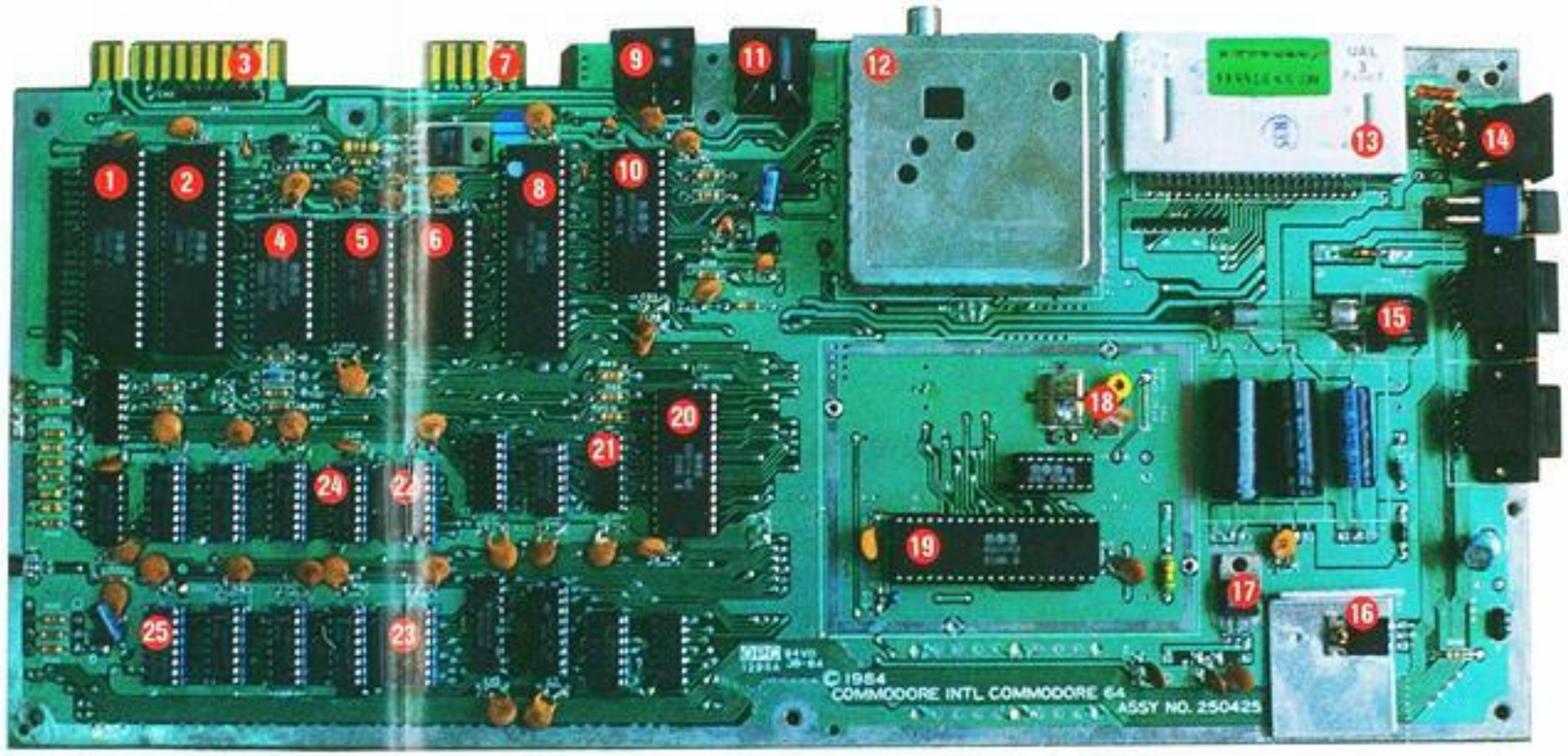
Şu yazıyı okuyan herkes en azından bilgisayarı çeşitli amaçlarla kullanıyor. Kimimiz hesabını bu aletlere emanet etmiş. Kimi yazışmalarını veya raporlarını bu aletle hazırlıyor. Kimimiz programlama amacıyla başına oturuyor. Kimileri de alet başında joystick eskitiyor.

Günümüzde hemen her amaçla kullanılan bu araçlara çoğu kimse daha yakından hitap edebilmek için özel programlama dilleri öğreniyor ve uyguluyor. Hemen her amaca uygun programlama dilleri tasarlanmış durumda. Mesela kolay kullanımlı ve güçlü diller olarak anılan BASIC ve PASCAL başı çekiyorlar. Bu iki dille kısa zamanda kendi amacınıza uygun programı geliştirip kullanmanız mümkün. Bazı başka diller ise daha detaylı, sistemli ve uzun süren çalışmalar gerektiriyor. Fakat gene de ASSEMBLER bile size aşılamaz dağlar gibi gelmesin. Fakat programlama konusunda ne kadar tecrübeniz olsa da bilgisayarın çalışması tam olarak ele alındığında birçok karanlık nokta içermektedir. 'Acaba şu kapalı kutunun içinde neler var?' veya 'Kara kara böceğe benzeyen bazı şeyler var, acaba bunlar ne yapıyor?', bir başkası 'O kadar şey yazıyorum hepsini aklında tutuyor ve doğru olarak yapıyor. Bunları nerede saklıyor?' gibi sorular çoğu kimsenin aklına takılıyor.

Dedik ki bu kadar soru var; bunları cevaplayacağız. Fakat insan birşeyin içini görmeden detaylı olarak incelemesi teorik ve yetersiz bir uğraşıdan ileri gitmez. Bunun yanında herkesin bilgisayarının içini açıp burada anlatılanları canlı olarak görmesini bekleyemeyiz. En basitinden birçok aletin garanti kapsamı dışında kalmasına yol açacaktır böyle bir olay. Bu yüzden biz sizin için bir COMMODORE 64'ü söktük, içinin fotoğrafını çektik ve tüm bölgeleri tek tek işaretledik. Burada anlattıklarımızı aynı zamanda bu şekilden izlemeye çalışın.

Bu yazıda her bir ünitenin adlarını tek tek belirtip sırayla hepsinin fonksiyonlarını açıklayacağız.

Bilgisayarımızın içi kara böceklere benzeyen, tümleşik devre diye adlandırılan elemanlarla doludur. Bunların hepsinin de tek tek özel isimleri vardır. Bunlar RAM, ROM, VIC, CPU, CIA, SID gibi veya benzerleridir.



Commodore 64'ün elemanlarının yerleşimi

1. CIA 1 (MOS 6526)
2. CIA 2 (MOS 6526)
3. USER-PORT
4. BASIC ROM'u
5. KERNAL ROM'u
6. Karakter Seti ROM'u
7. KASET PORT'u
8. ANA İŞLEM ÜNİTESİ (MOS 6510)
9. SERIAL PORT
10. PLA çipi
11. VIDEO çıkışı
12. Yüksek Frekans Modülatörü
13. EXPANSION PORT (Genişleme Yuvası)
14. Voltaj girişi
15. Akım doğrultucusu
16. Gerilim Regülatörü-1
17. Gerilim Regülatörü-2
18. Kuvars Kristal (Darbe üretici)
19. VIC çipi
20. SID çipi
21. Renk RAM'ı
22. Multiplexer-1 (Adres bitleri için)
23. Multiplexer-2 (Adres bitleri için)
24. 64 KByte RAM çipleri (dinamik tip, 8 adet)
25. Timer (RESET sinyalini üretiyor.)

Gelelim listemize:

ADRES: COMMODORE 64'ün hafıza kapasitesinin 64 kilobyte olduğunu hepimiz biliyoruz. Fakat bu tam olarak ne anlama geliyor?.. Şimdi bir sokak hayal edin ve bu sokağın üzerinde 65536 tane ev bulunsun, bundan başka her evin üzerinde sırasını belirten bir numara bulunsun. (Mesela 20111 numaralı evin hemen yanında 20112, onun yanında da 20113 numaralı evin bulunması gibi.) Böylece her evimizin bir adresi (numarası) ve içeriği (byte) olur. Sonuç olarak herhangi bir evin içeriğiyle bir iş yapmak istediğimizde adresi yardımıyla içeriğine erişebiliriz.

ADRES BUS'u: Bu kanal, bilgisayarın en önemli sinyallerinden olan adres (!) bilgisini taşır. Bilgisayarın ana işlem ünitesi (!) adres bus'una adresi yazar ve bu kanala bağlı diğer elemanlardan biri cevap verir; bu elemanlar RAM (!), ROM (!) gibi bir ünite olabilir. COMMODORE 64'ün bütün adres kapasitesini kullanabilmek için adres bus 16 tane ayrı hattan oluşmuş bir elektriksel bağlantı içerir. Bu 16 hattın herbiri bağımsız olarak '0' ve '1' durumlarından birde birinde bulunacağından 2↑16 = 65536 tane adres seçebilme ihtimalimiz var demektir. Bu sayı da zaten COMMODORE 64'ün adres kapasitesini yansıtır.

BASIC ROM'u (4): Bu rom'a aynı zamanda BASIC INTERPRETER (yorumlayıcı) rom'u da denir. Yorumlayıcının yaptığı iş o sırada kullanıcının emirlerini (BASIC emirleri) makinanın anlayacağı bir şekle çevirmek ve bunların yürütülmesini sağlamaktır. Bilgisayarımızın ana işlem ünitesi (!) yalnız makina dilinden anladığı için PRINT veya RUN gibi emirlerle doğrudan hiçbir şeye başlayamaz. Yüksek dillerdeki programlarımızı makinanın daha ilkel diline çevirip yürütülmesini sağlayan BASIC INTERPRETER ROM'undan başkası değildir.

BİT Bir bit bilgisayarın içinde rastlayabileceğimiz en küçük bilgi birimidir. İki durumu olabilir, bunlar '0' ve '1' durumlarıdır. Elektriksel olarak ise şase seviyesi ve +5V seviyesi olarak gerçekleşirler. Fakat bu kadarcık bilgi genelde kendi başına pek bir işe yaramaz. O yüzdendir ki çağdaş bilgisayarlar en az 8 bit üzerinden çalışacak biçimde yapılırlar. Bu demektir ki aletler artık bitle değil Byte'la çalışırlar (8 bit'lik bir bilgi gurubu bir Byte olarak anılır). Bir Byte'ın içeriği ise 256 değişik durum alabilir. Hafıza olarak bir tek byte pek bir şey ifade etmez. Birkaç bin byte bir araya geldiğinde hatırı sayılır bir hafıza bölgesi meydana getirebilirler. Hafıza ölçüsü olarak kullanılan başka bir birim de KByte'tır (Kilo-Byte). Bir KByte 1024 Byte'tan oluşmuştur. Bu tanımlara göre COMMODORE 64'ümüzün hafıza kapasitesi 64 KByte, yani 65536 Byte, yani 524288 Bit'tir.

CIA (1,2) (=Complex Interface Adapter=Giriş/Çıkış ara birimi): Eğer bilgisayarımız dış dünyayla bilgi alışverişi yapmasaydı, beş para etmez bir dökümtüden başka bir şey ifade etmeyecekti. CIA gibi veya benzeri ara birimler bu görevi üstlenen elemanlardır. Mesela aletimizin tuş takımı CIA denilen arabirim yardımıyla bilgisayarın ana devresine birleşmektedir. Bundan başka disk ünitemiz, kaset teybimiz, joystick'imiz ve User-Port'umuz hep bu üniteler vasıtasıyla bağlantı kurabilmişlerdir. Bütün bilgiler bilgisayara girip çıkmadan önce bu elemanlara uğrayıp değişik şekillere çevrilirler. Bu elemanlar ana işlem ünitesinin (!) adres bölgesinde belli bir yer kaplarlar ve ana işlem ünitesi bu bölgelere birşey yazdığında bu bilgi elektriksel sinyaller olarak bilgisayar devresinin dışına çıkarlar; aynı şekilde dışarıdaki elektriksel sinyaller ana işlem ünitesi tarafından bu adreslerin okunması vasıtasıyla algılanır. Başta da söylediğimiz gibi bu elemanlar olmasaydı bilgisayarımız kör ve sağır kalacaktı.

ÇEVRESEL BİRİMLER: Bkz. peripherie.

DATA BUS'u (Bilgi kanalı): Data bus'u, ana işlem ünitesinin adres bus'u yardımıyla seçtiği Byte'ın okunmasını veya yazılmasını mümkün kılar. Bilginin akış yönünü belirleyen sinyal ise R/W hattı tarafından taşınır. Dikkat ettiğiniz gibi data bus'u bir Byte'lık bilgiyi paralel olarak (!) taşımak için 8 kanallı bir hat olmak zorundadır.

DATA MEDIUM (Bilgi saklama aracı): Bilindiği gibi bilgisayarı kapattığımızda hafızası tümüyle silinmektedir. Programlarımızın kalıcı olmasını sağlamak için bu bilgileri aktaracağımız bir araca ihtiyacımız olacaktır. En ilkel örneğiyle bu işi delikli kartlar yapmaktadır. Fakat günümüzde bu çok eski teknoloji terkedilmiştir. Biz genelde bu iş için manyetik kayıt araçlarını kullanıyoruz (disket ve kaset gibi).

Hep duyarız: Hardware... eprom... interrupt... data bus... Bunların aslında Commodore 64'ün çalışmasını anlatan önemli kavramlar olduğunu biliyor muyuz acaba?

EPROM: (Erasable Prog Rammable Read Only Memory): Eprom'larda bir tür kalıcı kayıt araçlarıdır. Tümleşik devre olarak imal edilmişlerdir. Bu yüzden görev olarak ROM'larla eşdeğerdir. Fakat yapı olarak ROM'lara göre önemli bir üstünlükleri vardır; o da silinebilir olmalarıdır. EPROM'ların silinme işlemi kuvvetli ultraviyole ışını altında 10-30 dakika bekletmek suretiyle olur. Böylece normal çalışma ortamı altında hiçbir tehlike arzetmezler. Eprom'ların tekrar tekrar silinip programlanabilme özelliği onları bilhassa araştırma geliştirme işlemlerinin vazgeçilmez unsurları haline getiriyor. Şayet aletinizde işletim sistemini (!) veya karakter setini (!) kalıcı olarak değiştirmek istiyorsanız en rahat EPROM'larla bu işin üstesinden gelebilirsiniz.

EXPANSION-PORT (13): COMMODORE 64'ün en önemli özelliklerinden biri expansion-port'u bulunmasıdır. Bu port'ta bilgisayar sisteminin en önemli sinyalleri yer almaktadır. Böylece aletinizin içini açmadan, içindeki birçok işlemin akışını değiştirmek mümkün olmaktadır. Bu port'un kullanımına en basit örnek olarak kartuşları verebiliriz.

HARDWARE: Bilgisayar sistemini oluşturan tüm parçacıklar. Tümleşik devreler, transistörler, rezistanslar, diyotlar, kapasitörler, switch'ler, teller, plaket v.s. Kısacası aletimizin maddi olarak algılanabilen tüm elemanları. Bu elemanların hepsi sabit olarak oluşturulmuş durumdadır ve değiştirilemez.

INTERRUPT: Bilgisayarın düzenli olarak çalışması için gerekli olan bir işlem. COMMODORE 64 saniyede 60 kere bu işlemi yapmak zorundadır. Yapılan işlem şudur: O sırada yapılan işlem kesilir ve işlemin o anki durumu ile bilgiler bir yere saklanır; ve interrupt altprogramı yürütülür. Bu altprogram aletimizin işletim sistem rom'unda (!) bulunmaktadır. Interrupt sırasında bazı önemli fonksiyonlar yerine getirilir. Bunlar klavye ve joystick'lerin sinyallerinin değerlendirilmesi, sistem saatinin 1/60 saniye ilerletilmesi gibi işlemlerdir. Bu işlemler bittikten sonra kesilmiş olan program kaldığı yerden devam ettirilir. Interrupt işlemi periyodik olarak yürüyen bir işlem olduğu için daha değişik amaçlarla da kullanılabilir. Mesela, başka programlar çalışırken arka planda 3 kanaldan hareketli bir müzik parçasının çalınması gibi. Tabii bu ve bunun gibi işlemleri gerçekleştirmek sanıldığı kadar kolay bir iş değildir. Böyle programları yazabilmek normal programlamadan ayrılıp apayrı bir programlama tekniği olarak anılmaktadır (interrupt programlama tekniği).

KARAKTER SETİ ROM'U (5): Bu eleman sabit bir bellek tipi olup içeriği ekranda görülebilecek tüm karakterlerin nokta nokta tanımıdır. Bu tanımları değiştirip EPROM'a programlarsanız böylece kendinize ait kalıcı bir karakter setine sahip olursunuz.

KERNAL-ROM (5): Bilgisayarın zekâsı. Aynı zamanda 'işletim sistemi' olarak da anılır. Yaptığı işlem ise sistemi çalışır vaziyette tutmak için bütün gerekli verileri saklamak ve bunları işlemektir. Genelde tüm çevresel işlemleri de işletim sistemi üstlenir. Böylece ekrana nasıl yazı yazacağız, veya çevresel araçlarla (!) nasıl komünikasyon kuracağız diye tek tek uğraşmamız gerekmemektedir; işletim sisteminde bütün bu işleri yapacak altprogramlar mevcuttur. Böylece programcı detaylarla yorulmayacak ve programının daha genel akışıyla, mantığıyla v.s.ilgilenecektir.

MODUL: Bilgisayara bir program yerleştirmenin en basit yolları, programı herhangi bir data medium'undan (kaset veya disket) geri çağırmak veya programı tuş takımından alete girmektir. Bunlardan başka kullanılan bir yöntem de programı eproma (!) "yakmak" ve bu epromu kartuş biçiminde bilgisayarın hafızasına dahil etmektir. Eprom'un içeriği aletin hafızasına dahil olduğundan artık yükleme diye bir sorunumuz olmayacaktır, çünkü programımız zaten aletin hafızasındadır. Modüllerin bilgisayarla birleştiği yer expansion-port'tur.

PARALEL KOMUNİKASYON: COMMODORE 64'ümüzün 8 bit'lik paralel çalışan bir data bus'u vardır. Böylece ana işlem ünitesi bir kerede bir Byte'ı olduğu gibi okuyabilir veya yazabilir. Paralel bilgi taşıma yöntemi genelde hızlı bir yöntemdir. Bundan başka seri bilgi taşıma (!) yöntemi vardır ki bu yöntem daha yavaş kalmaktadır.

PERIPHERIE (Çevresel birimler): Bilgisayara dışarıdan bağlanan ve bilgisayarın standart üniteleri arasında yer almayan elemanlar. Mesela disksürücü, bilgiyazar, joystick ve hatta monitör ekranı. Personal Computer diye adlandırılan bir üst sınıf bilgisayarlarda tuş takımının bile periphe-

rie sayıldığı olur (Mesela AMIGA'da tuş takımı kendi başına bir bilgisayardır ve kendi ana işlem ünitesi vardır).

PLA (10) (Program Logic Array = Hafıza işlemleri koordinatörü; COMMODORE 64'teki görevi): COMMODORE 64'ümüzün ana işlem ünitesi yalnızca 64 KByte adresleyebiliyordu. Fakat C-64'ümüzde daha 20 KByte hafıza elemanı bulunmaktadır. Bu 20 KByte, 8 KByte'lık INTERPRETER (!), 8 Kbyte'lık KERNAL (!) ve 4 KByte'lık KARAKTER SETİ'nden oluşmuştur. Bu toplam 84 KByte'ın ana işlem ünitesi tarafından kullanılabilmesini sağlamak için tasarımcılar bir hile bulmuşlar; aletin tüm hafıza alanını kaplayan RAM'in bir kısmını ROM'la kaplamışlar. Böylece ana işlem ünitesi o sırada seçilmiş konfigürasyona bağlı olarak RAM'e veya ROM'a ulaşmakta. Bank Switching diye de adlandırılan bu işlem PLA'nın asıl görevi.

POINTER (Göstergeç): Yazdığınız program yürürken, bazı değişkenleri veya text'leri bir yerlerde saklamak zorundadır. Bunlar, çalışma alanının belirlenmiş bazı yerlerine yazılarak saklanır. Göstergeç bu bilgileri saklamak için ayrılmış bölgenin başı ve sonu ile ilgili bilgileri içerir. Böylece saklanmış olan bilgileri tekrar bulmak mümkün olur.

PROCESSOR (CPU = Central Processing Unit = Ana İşlem Birimi) (8): Bilgisayarımızın beyni olan ünite. Her türlü mantıksal ve matematiksel işlem bu ünitede yapılır. Hafıza ile ilgili birçok işlem de (bir bölgeyi kaydırma veya kopya etme) ana işlem ünitesinin bazı özel register'leri (!) yardımıyla yapılır. Bütün sistemi kontrol eden bir orkestra şefine benzetebiliriz o'nu. Bilgisayarın hangi elemanının hangi işlemi yapacağına karar verip onlara emirler gönderen de ana işlem ünitesidir.

QUARTZ (18): Bütün bir bilgisayar sisteminin işini kusursuz olarak yapabilmesi için tüm elemanların işlerini belli bir senkronizasyon modeline göre yapmaları gerekmektedir. Bildiğiniz gibi aletimizin bir adres bus'u ve bir data bus'u vardır; bu yüzden bu bus'ların kullanımı sırasında zamanlamadan doğacak bir çakışma (hiç kuşkunuz olmasın) zincirleme bir yanlışlıklar reaksiyonunu başlatacak ve aletimiz elden gidecektir. Tüm elemanların belli bir ritm içinde çalışmasını sağlayan şey sistem darbeleridir. Tüm elemanlar hangi darbe sırasında bus'ları kullanacaklarını bildiklerinden hiçbir çakışma olmayacaktır. COMMODORE 64'ün PAL tipi (Türkiye'de de kullanılan sistem) 0.98 MHz'de çalışmaktadır. Bu demektir ki sistem saniyede 980000 darbe üretmekte. Aslında bu hız günümüz teknolojisiyle kıyaslandığında oldukça düşük kalıyor. Mesela Personal Computer'ler 4.7 MHz'de çalışır. AMIGA ise 8 MHz gibi bir çalışma hızına sahip.

RAM (24) (Random Acces Memory = Rastgele ulaşımlı bellek = okunabilir/yazılabilir bellek): COMMODORE 64'ün 64 KByte RAM belleği vardır. Tam olarak 65536 Byte'e karşılık gelen bu değer aynı zamanda 65536 karakterlik bir text'i hatırında tutabilecek bir kapasite. RAM'ler ana işlem ünitesine adres bus ve data bus'u yardımıyla bağlıdır. Bilginin akış yönünü ise R/W sinyali belirler. Bilgisayarın çalışma alanı böyle bir bellekle oluşturulduğu için bir bilgisayarın RAM hafızasını artırırsanız, aynı zamanda işlem yapma gücünü de artırırsınız. Kısacası RAM bilgisayarın içindeki en önemli birimlerden biridir.

REGİSTER: Bilgisayarın içindeki birçok eleman kendilerine ne iş yaptırılmak istendiğini bilmek gereksinimindedirler. Bu da ana işlem ünitesi ile söz konusu elemanın ortak olarak kullanılabileceği RAM tipinde belli bir bellek alanı gerektirir. Bu bölgeler elemanın yapacağı işle ilgili bilgileri içerir.

İkinci tip register'ler ise ana işlem ünitesinin kendi içinde bulunur. Bunlar da RAM özelliği taşıyan birkaç Byte'lık bellek bölgeleridir. Bu register'lerin çok çok özelleşmiş ve çok çok önemli görevleri vardır, çünkü ana işlem ünitesinin işlem yapabilmesini sağlayan bu register'lerdir. Bu register'lere örnek olarak Akkü register'ini ve program sayacını gösterebiliriz.

RESET: Bu işlem de bilgisayar için önemli bir yer işgal ediyor. Yapılan işlem bilgisayarı meydana getiren elemanları daha evvel belirlenmiş bazı değerlere getirmekten ibarettir. Bu işi yapan ise işletim sistem rom'unda (!) bulunan reset altprogramıdır. Reset işlemi genelde alet açıldığında otomatik olarak yapılır. Tabii bundan başka kullanıcının isteği üzerine de gerçekleştirilebilir. Diyelim ki reset-switch'iniz var ve aletiniz "crash" etti; aleti kapayıp açacak yerde reset tuşuna bir kere dokunursunuz.

ROM (Read Only Memory = yalnızca okunabilir bellek) (4,5,6): Bilgisayar aslında o kadar da akıllı bir alet değildir. O'na ne yapacağının daha evvel söylenmiş olması gereklidir ki bir şeyler yapabilsin. Bu yüzden bilgisayarın içinde her zaman bulunan ve silinip, değiştirilmesi imkansız, sabit programlar bulunur. Bu programlar ROM denilen tümleşik devrelerde bulunur. Bu elemanların başlıca özelliği sabit içerikli bellek üniteleri olmalarıdır. COMMODORE 64'ün içinde üç tane ROM bulunmaktadır. Bunlar INTERPRETER ROM'u (!), KERNAL ROM'u (!) ve KARAKTER SETİ ROM'udur (!).

RAM. Hani bilgisayar alınmaya gidildiğinde "Kaç K acaba?" diye sorarlar ya... İşte bu, RAM'ın büyüklüğüne ilişkin bir sorudur. Peki RAM ne iş yapar ve neden bu kadar önemlidir?

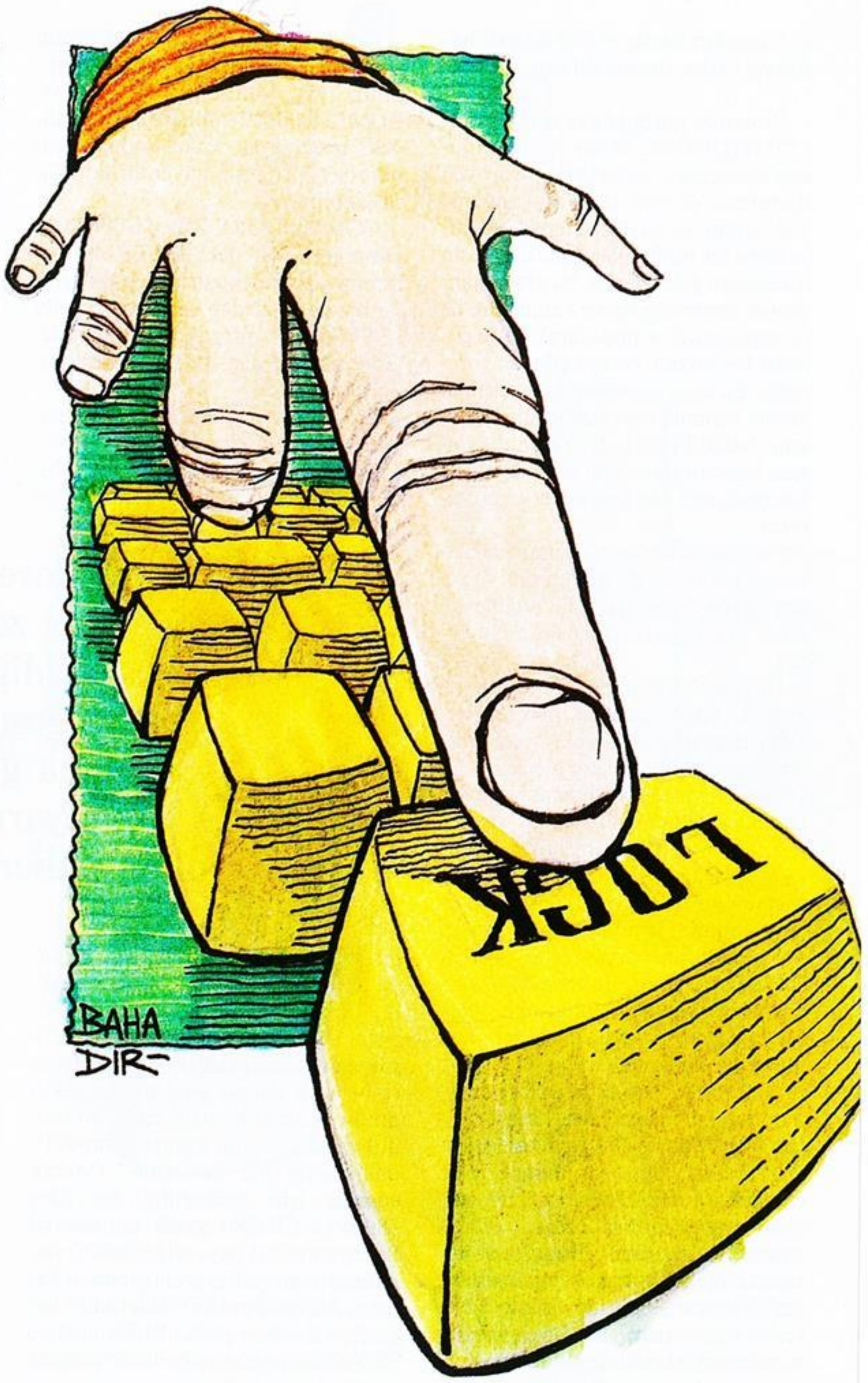
Aynı zamanda işletim sistemini oluşturan bu üç rom'un görevi, sistemin tüm elemanlarıyla kullanıcının bağını kurmak ve kullanıcının emirlerini mümkün olan en az sıkıntıyı vererek (makine dili yerine BASIC'te programlama) yürütmektir.

SERİ KOMUNİKASYON: Bu haberleşme tipinde COMMODORE 64 bilgi alışverişi için seri port'u kullanır. Bu port'un özelliği haberleşme için yalnızca tek hat içermesidir. Yani bir Byte'lik bilgi gönderecek olursak bu bilgi bitlerine ayrılacak ve öyle gönderilecektir. Görüldüğü gibi bu işlem daha fazla zamana mal oluyor. Fakat bunun yanında avantaj olarak sekiz hat yerine tek hat kullanmamızı gösterebiliriz.

SID (20) (Sound Interface Device = Müzik çipi): Her türlü sesi üretme becerisine sahiptir. Yani COMMODORE 64'ümüz komple bir ses synthsizer'ı içermektedir.

TIMER (25): Bu eleman, aleti ilk açtığımız zamanki kargaşayı düzenliyor. Bilgisayarın tüm elemanları yeni uykudan kalktığı için bunların tümüyle uyanmaları için belli bir süre beklemek gerekiyor. Yoksa uyanır uyanmaz bu elemanlardan bir iş beklersek büyük ihtimalle işlerini karıştıracak ve aletimiz genelde uyanamayacaktır. Daha evvel de belirttiğimiz gibi bilgisayarın doğru olarak çalışması için kusursuz bir uyum içinde olması gerekmektedir.

TUŞ TAKIMI: Şayet her tuşa ayrı bir hat çekip bunları bilgisayarın plaketinde arabirimlere verseydik; 67 hatta ve 67 bitlik arabirim kanalına ihtiyacımız olacaktı. Böyle bir yapı hem gereksiz, hem de pahalı olduğu için tuşlar matris biçiminde bağlanmıştır. Böylece 16 hat ve arabirim kanalıyla aynı iş görülür. Bu bağlantının detaylarını şekilde görmektesiniz.



USER-PORT (3): Burası seri-port gibi çevresel birimlerin kontrolunu sağladığı gibi çeşitli elektriksel kumanda aletlerini kontrol edebilir. Tabii böyle bir işi özel yazılmış bir program becerebilir. User-port dışarıya elektriksel sinyaller gönderebildiği gibi aynı zamanda dış çevreden gelen elektriksel sinyalleri algılayabilir. Böylece mesela dışarıdan algılanan değerleri bir program yardımıyla analiz edip değerlendirebiliriz.

VIC (19): (Video Interface Controller = Görsel arabirim aracı): Bu araç olmasaydı, herhangi bir monitörün veya televizyonun ekranında bilgisayarımızın yaptığı işleri göremeyecektik. Bu aracın yaptığı iş ekran hafızasından (RAM'in bu iş için ayrılmış bir bölgesi) karakter kodlarını okuyup karakter seti rom'undan (!) her karakterin şeklini alıp bu şekli ekranı çizen elektron ışınına kumanda edip görünür hale getirmektir.

ZEROPAGE: Bu hafıza bölgesi ana işlem ünitesine ve işletim sistemine ait çok önemli ara değerleri içermektedir. Aslında bu bilgiler RAM'-

in başından başlayıp 820 adresli hafızaya kadar uzanmaktadır.

Yukarıda gördüğünüz açıklamalar COMMODORE 64'ün içinde bulunan elemanları, sinyalleri ve yürüyen işlemlerin önemli olanlarını anlatıyor. Şimdi ise bu ufak eleman ve sinyallerin bir bütün olarak nasıl bir işin üstesinden geldiklerini, bu arada hangisinin devreye girip ne kadar süre ile ne yaptığını ve sonuç olarak gözle görülür bir işlemin ortaya çıkışını göreceğiz. Bu anlatım sırasında da eleman adının yanında parantez içinde bir rakam bulabilirsiniz. Bu rakam yukarıda bahsettiğimiz gibi hâlâ elemanın fotoğrafımız üzerindeki yerini gösteriyor.

Fonksiyonları en başından ele alalım ve konuya öyle yaklaşalım. Diyelim bilgisayarın ON/OFF düğmesini açtık, bilgisayarın içinde neler olmakta?..

İlk olarak bütün elemanlar şok geçirecek kadar çok akım çekerler. Tüm elemanların hayata geçebilmeleri için sakinleşmeleri beklenir. Bu yüzden hemen tüm elemanlara uygulanan RESET sinyali belli bir süre şase seviyesinde kalır. Bu sinyal bilhassa bilgisayarımızın beyni sayılan ana işlem ünitesinin uyanıp çalışır vaziyete gelebilmesi için hayati önem taşır.

Sistemin bir parçası olan TİMER (25) devresi yalnız bu anda aktif olarak görev yapar. Yaptığı iş ise belli bir zaman (bilgisayarın içindeki voltaj seviyesi normal seviyesine ulaşıp tüm devreler aşırı akım çekme durumundan kurtulana kadar) beklemektir. Daha sonra tüm elemanları serbest bırakır, ana işlem ünitesi de bu elemanlara dahildir.

Bilgisayarımız donanım olarak uyandı ve şu anda çalışabilecek durumda olan bir ana işlem ünitemiz var. Fakat o şimdi ne yapacak? Nereden başlayacak? İşte böyle bir durumda ana işlem ünitesinin içgüdüsel hareketinden söz edebiliriz. Ana işlem ünitesi RESET işlemine her defasında aynı şekilde başlaması gerektiğinden bu çip üretilirken içine sabit bir emir konur ve bu emir yalnızca böyle bir durumda kullanılır. COMMODORE 64'ümüzün ana işlem ünitesi 6502/6510 tipindedir ve bu aileye ait ana işlem ünitelerinin hepsinin standart başlama noktası $FFFC (65532)'deki göstergecin gösterdiği adrestir.

Gerek bu göstergeç, gerek onun gösterdiği adres, işletim sistemi rom'unun (5) içindedir. KERNAL ROM'u da denilen bu bölgenin içindeki programlar kalıcı olduğu için buradaki program güvenilirlikle yürütülebilir.

COMMODORE 64'te $FFFC'deki göstergeç 64738'deki RESET altprogramını göstermektedir. Bu altprogramı biz de dışarıdan çağırabiliriz. Bu işi SYS64738 yazarak yapabilirsiniz. Göreceksiniz alet ilk açılış durumuna dönecektir.

İlk önce ana işlem birimi için hayati önemdeki işlemler yürütülür.

Bunlar ana işlem ünitesinin çalışma modunu belirlemek ve içindeki yığın göstergecini sıfırlamaktır (aslında sıfırlamanın tam tersi yapılmakta, çünkü 6502/6510 ana işlem ünitesinin yığını yukarıdan aşağıya prensibiyle çalışmaktadır). Bundan sonra belli bir şifreye göre modul takılı olup olmadığı kontrol edilir ve modul varsa sistemin kontrolu modul'daki programa devredilir. (Merak edenler için belirtelim, bu şifre $8004'te CBM80 yazılı olmasıdır). Şayet modul yoksa diğer aktif elemanların inisyalizasyon işlemleri yapılır. Bu, şu demektir: Bazı sabit değerler CIA'lere (1,2), VIC'e (19) ve SID'e (20) yazılır ve belli bir çalışma disiplinine sokulurlar.

Daha sonra yapılan işlem ise RESET sırasında en fazla zaman alan işlemdir. Bu işlem RAM'in (24) test edilmesi ve çalışma alanı olarak daimi ve kesiksiz ne kadar hafızanın bulunduğunun tespit edilmesidir. Bundan sonra ise RAM'in işletim sistemi tarafından kullanılacak bölgesi ROM'dan bazı değerlerin kopya edilmesiyle ilk açılış değerlerine getirilir.

Bu işlemler yerine getirildikten sonra ekran için ayrılmış olan hafıza bölgesi silinir. Burada RESET işleminin işletim sistemi ile ilgili kısmı bitmiştir. Daha sonra ana işlem ünitesi $A000 (40960)'da bulunan göstergeç yardımıyla BASIC-WARMSTART'a gider. (Burada ise bildiğimiz BASIC INTERPRETER ROM'u başlamaktadır.) Ana işlem ünitesi artık INTERPRETER (yorumlayıcı) döngüsüne girmiştir; artık bizden BASIC'te emir alabilir ve yürütebilir. Bilgisayarın bu duruma geldiğini şuradan anlarız: ekran görünür vaziyette açılmıştır ve açılış mesajı görünmektedir, CURSOR yanıp sönmektedir ve tuş takımından yazdıklarımızı alabilmektedir.

Commodore 64'ünüzü açtınız ve mavi zemin üzerinde alışageldiğiniz yazıların oluştuğunu gördünüz. Bu duruma gelmek için bilgisayarınızın içinde neler oldu, haberiniz var mı?

Artık bilgisayarımız uyanmış ve günlük yaşamına dönmüştür. Artık bizden emirler alabilir ve bunları uygulayabilir. Şimdi iş daha da ilginçleşiyor. Acaba bilgisayar bizim istediklerimizi yerine getirirken ara işlemler olarak bize göstermeden neler karıştırıyor?

Diyelim ki CURSOR'umuz bir bölgede yanıp sönüyor. Ve biz bir tuşa bastık. CURSOR bastığımız tuşa ait karakteri yazarak bir ilerliyor. Tam detayları gözönünde tutulduğunda acaba bu olay nasıl gerçekleşiyor?..

Önce tuş takımını inceleyerek başlayalım. Tuş takımındaki her tuş elektriksel bir anahtar vazifesi görmektedir. Biz bir tuşa bastığımızda elektriksel devreyi kapatırız ve bilgisayar arabirim ünitesi yardımıyla o tuşa bastığımızı algılar. Bu sefer başka bir sorun çıkıyor. COMMODORE 64'ün üzerinde tamı tamına 66 tane tuş bulunmakta. Bu 66 tuşun bilgisayar tarafından algılanabilmesi için 66 hat artı bir adet de ortak şase sinyali için bir hat, yani toplam 67

hat gerekiyor. Tabii üretim sırasında 67 kablonun yaratacağı sorunları siz düşünün.

Böyle bir sorunun farkına varan bilgisayar tasarımcıları matris bağlantısı adında bir yöntem önermişler. RESTORE tuşu dışında kalan tuşları 64 tuş olarak görebiliriz, çünkü SHIFT LOCK tuşu soldaki SHIFT tuşu ile paralel bağlıdır. Böylece bu 64 tuşu 8 × 8'lik bir matrise bağlayabiliriz. RESTORE tuşu oldukça özel bir tuş olduğundan yalnızca kendine özel iki hatta sahiptir. Böylece tuş takımından bilgisayar plaketine yalnızca 18 hat taşıyarak bütün tuş takımının bilgisayar tarafından algılanmasını sağlayabiliriz. Bundan başka kullanılması gereken arabirim ünitesi kanal sayısı ise 64'ten 16'ya inmektedir.

Tuş takımından gelen sinyaller CIA'lere girer. İşletim sisteminde matris tipine uygun bir altprogram, hangi tuşun basıldığını belirler. Bu programın dikkat etmesi gereken başka bir husus da tuşların tek mi, (SHIFT)'le birlikte mi, (C=) ile birlikte mi, yoksa (CTRL) tuşu ile mi basıldığıdır. Çözüm şöyle bulunmuştur: Her tuşun belli bir numarası vardır. Fakat bu numara kendi başına hiçbir işe yaramaz; bu rakamın basılan tuş kombinasyonuna göre belli bir ASCII koduna çevrilmesi gerekmektedir. Bu çevirme işlemleri için işletim sistem rom'unda dört tane (her kombinasyon için bir tane) deşifre tabloları bulunmaktadır. Böylece basılan tuş kombinasyonuna göre hangi ASCII karakteri gerekiyorsa elde edilir.

Buraya kadar işletim sistemi hangi tuşa bastığımızı algıladı ve istediğimiz karakteri ekrana basması gerekiyor. Önce bu ASCII kodunu ekran koduna çevirir ve 1024'ten 2023'e kadar adresler arasında uzanan ekran hafızasının belli bir yerine yazar. VIC çipi ise ekran hafızasından bu kodu alır ve KARAKTER SETİ ROM'undan (6) bu karakterin şekil olarak tanımını alır ve elektron tabancasına kontrol ederek bu karakterin görünür olmasını sağlar. Gayet düzenli ve birçok işin yapılması gerekiyor; yalnızca tek karakterin görünür hale gelmesi için.

İlgi çekebilecek başka bir işlem de bilgisayarın disketten bir dosyayı okumasıdır. Görülüyor ki işletim sistemine burada da birçok iş düşmekte.

Diyelim emri verdiniz. Bu emrin doğruluğu kontrol edildi, analizi yapıldı ve artık bilgisayar tam olarak ne istediğinizi biliyor. Artık sıra bilgisayarın kolları sıvayıp işe girişmesine gelmiştir. Ana işlem ünitesi, işletim sistemindeki bu işle ilgili bir altprograma gider. Bu altprogramda yapılan işlem arabirim ünitelerini aracı olarak kullanmak suretiyle disk sürücüsüne bir emir kanalı açar ve istenilen tipte ve addaki dosyayı hazır etmesini emreder. Bu arada emre gelen cevabı değerlendirerek dosyanın başarılı bir biçimde açılıp açılmadığını kontrol eder. Dosya başarılı olarak açılmışsa göstergeçleri kullanarak dosyayı hafızaya okur. Bunu şöyle gerçekleştirir: Arabirim üniteleri vasıtasıyla tek bir Byte'ı alır. Şimdi bu Byte'ı bir göstergecin gösterdiği adrese yazmak gereklidir. Adres bus'una bu adres'i, data bus'una aldığı Byte'ı gönderir; bu arada R/W sinyalini yazma durumuna geçirir. Böylece dosya Byte Byte okunup hafızaya alınır. Dosyanın sonuna gelindiğinde ise disk sürücüsüne dosyayı kapatması için emir gönderir. Böylece dosyayı okuma işlemi biter ve ana işlem ünitesi diğer emirleri yürütmek için yorumlayıcıya geri döner.

Sonuç olarak görüyoruz ki işletim sistemi bizim için çok sıkıcı ve hassas işleri üstleniyor. Genelde işletim sistemlerinin yapıları da budur. O bilgisayarın standart ve en gerekli, en çok kullanılan işlemlerini üstlenirler. Bu işlemler genelde o bilgisayarın genel özelliklerini belirler.

Yapılan işlemlere son bir örnek verelim: Diyelim ki tuş takımı vasıtasıyla bir satır yazdık ve (RETURN) tuşuna bastık. Neler olmakta?..

Siz RETURN'e basar basmaz, işletim sistemi bu satırı alır ve $0200 (512)'den itibaren olan bölgeye taşır (kopya eder). Satırın sonunda ise satır sonunu belirtmek için içeriği '0' (sıfır) olan bir Byte eklenir. Yorumlayıcı ilk karakteri kontrol eder. Bu karakterin rakam olup olmaması çok şey değiştirir çünkü.

Eğer ilk karakter bir rakam ise, bu satır BASIC programının içine eklenecek demektir. Önce daha evvel aynı satırla yazılmış bir satır varsa o silinir ve programın geri kalan kısmı geri kaydırılır. Bundan sonra yeni satır yerine konur. Bu işlem ise yeni satırın gireceği yerden gerisini o satırın büyüklüğü kadar ileri kaydırıp açılan boş bölgeye o satırı kopya etmekle olur.

Eğer ilk karakter bir rakam değil ise yorumlayıcı (4) büyük bir tercüme tablosundan o emre ait altprogramın adresini arar. Bu altprogram daha evvel yazılmış, makina dili yapıda bir programdır ve yaptığı iş ise karşılık gelen BASIC emrinin yapması gereken tüm işlemlerdir. Bu emirin uygulaması bittikten sonra ana işlem ünitesi yorumlayıcı döngüsüne geri döner ve yeni emirler için hazırdır.

Bu makalede genel olarak bir bilgisayar sisteminin çalışması için gerekli üniteleri ve bu ünitelerin ortak olarak bir düzen içinde nasıl işlediklerini anlatmaya çalıştık. Bazı yerlerde detaylara da girdik. İsterdik ki bütün prensipleri ve detayları içerebilelim. Fakat bunun için ne bizim yerimiz, ne de sizin zamanınız yeter. Biz burada aklınıza takılmış olabilecek bazı konulara ışık tutarak bu konularda araştırmacı olmanız konusunda cesaretlendirmek istedik. Aslında sizin işiniz bundan sonra başlıyor. Burada değinilen konuların ışığında belli spesifik konuları ayırt edecek ve bunlar arasında seçim yapabilecek kadar bilginiz var artık. Yaptığınız seçim doğrultusunda bilginizi ve tecrübenizi artırın; daha sonra da konunuzun kapsamını genişletin. Göreceksiniz ki belli bir yerden sonra aşağı yukarı her konuda belli bir bilginiz olacak. Bu arada bir tavsiyemiz var: Bilgi ve tecrübenizi sistemli bir biçimde geliştirin. Yoksa bir yerden sonra kaybolursunuz. Ayrıca bilmediğiniz bir konuya ani olarak yönelmeniz gerekirse daha evvelden elde ettiğiniz sistematik çalışma becerisi hem kolaylık sağlayacak, hem de zamandan kazanacaksınız.

Son olarak bu makalenin beraberindeki 'bilgisayarın blok şeması' şekli üzerinde birkaç şey söyleyelim. Şu ana kadar elde ettiğiniz bilgilerle bu şema üzerinde tüm elemanların birlikte çalışmasını daha iyi izleyebilirsiniz.

Teleteknik'ten

Teleteknik Genel Kurulu yapıldı

Commodore firmasının Türkiye distribütörlüğünü yapan Teleteknik Elektronik Sanayi ve Ticaret A.Ş.'nin olağan genel kurulu 15.4.1987'de yapıldı. Geçen döneme ilişkin çalışma raporlarının incelenmesinden sonra yeni döneme ilişkin kararların gözden geçirildiği genel kurulda şirket yönetimine Osman Kavala, Zeki Türkkan, Bilal Sadi Oğuz, Rıza Çebi ve Ömer Erkmen seçildiler. Yeni Yönetim Kurulu önümüzdeki günlerde ilk kez toplanarak görev dağılımı yapacak. Yeni Yönetim Kurulu'na önümüzdeki çalışma döneminde başarılar diliyoruz.

Commodore, Bilgisayar Günleri'ndeydi

Günler çoğalıyor... Bu sefer de Robert Lisesi, öğrenciler arasında geniş ilgi toplayan bir Bilgisayar Günleri düzenledi. 6-10 Nisan'da Commodore, Apricot, Amstrad, Atari ve Apple bilgisayarları Robert Lisesi'nde kendilerini tanıttılar. Ama yalnız Commodore bunu bir geniş video ekranı kullanarak yaptı. Video gösterimlerinde bile fazla yaygın olarak kullanılmayan bir gerecin bir bilgisayar ekranı olarak kullanılıyor olması, Commodore pavyonunun önünde izdiham yarattı. Ayrıca Commodore artık bu tür fuarlarda geleneksel hale gelmiş olan çekilişlerinden birinde çeşitli hediyeler dağıttı. 7.4.1987'de Robert Lisesi'nde yapılan çekilişte kazananlar ve hediyeleri şöyle:

C-16: Ali Yeteroğlu *(Avusturya Lisesi).*

Işıklı kalem: Netsi Hullu *(Özel Dost Lisesi),* Meltem Mısırcı *(Avusturya Lisesi),* Ali Gökmen *(Avusturya Lisesi).*

Dergi seti: Nahoko Ando *(I.I.C.S.-Ommunity School),* Selim Ak *(Özel Dost Lisesi),* Gökhan Erkıralp *(Robert Lisesi),* Orkun Sak *(Robert Lisesi),* Basat Tayfun *(Robert Lisesi).*

Hata Köşesi

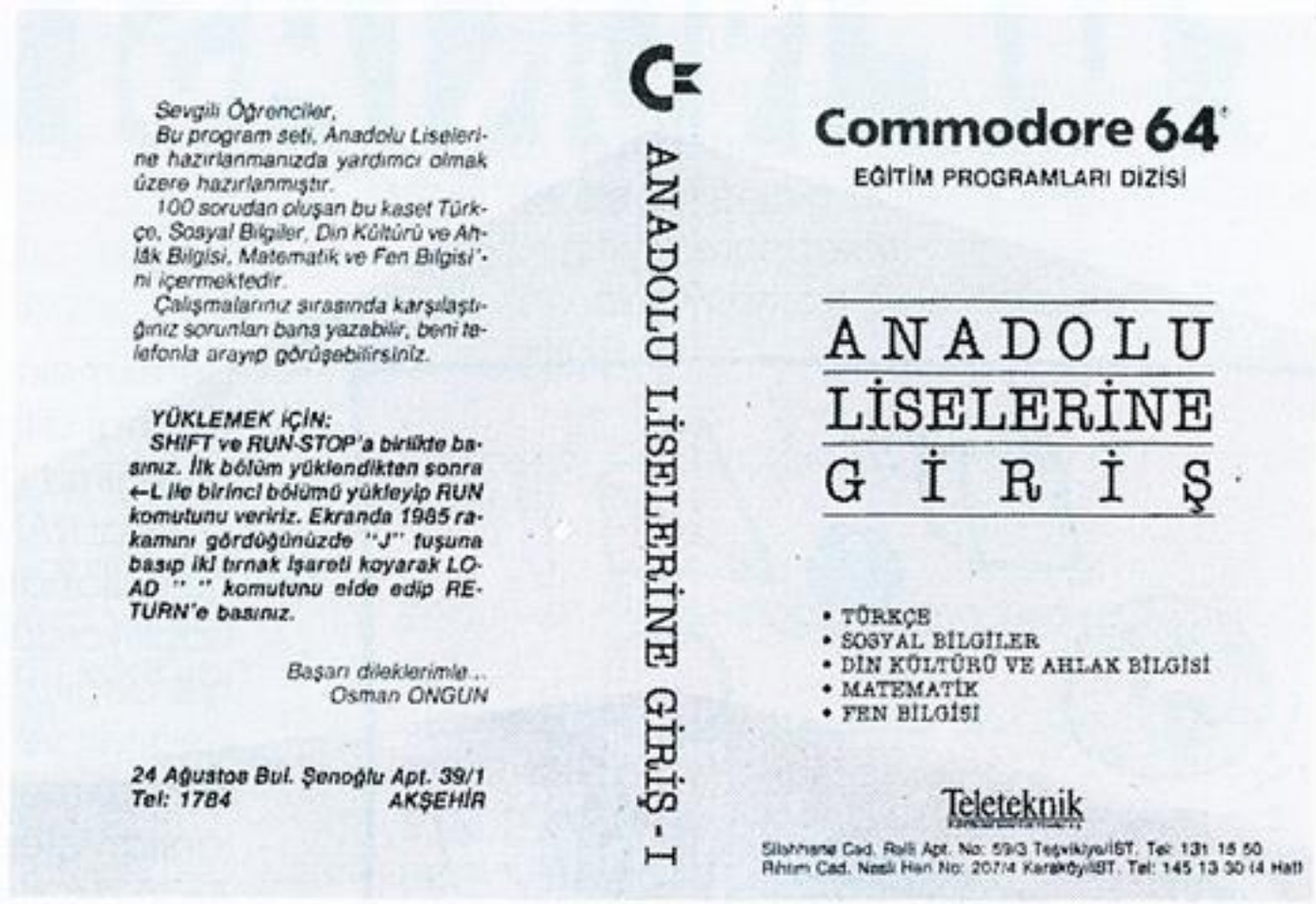
Korkarız, bu ayki hata köşemizde itiraf etmek zorunda olduğumuz en büyük ayıp, beş okurumuzu hayal kırıklığına uğratmak olacak. 14. sayımızın 50. sayfasında yayınlanan *Commodore Show* çekiliş sonuçlarında hata yapılmış. 16 Şubat çekilişinde aslında birer ışıklı kalem kazanan şu okurlarımız, yanlış olarak birer MPS 802 Printer kazanmış gibi gösterilmiş: Uğur Aslanalp, Ali Dikmen, Rafet Gürel, Serkan Özdemir, Neşe Gözdemir.
Diğer yandan gerçekten MPS 802 kazanan okurumuzun ismi ilan edilmemiş. Evet, 16 Şubat günkü çekilişte *Kenan Keleş* bir adet MPS 802 kazandı. 21 Şubat günkü çekilişte ise birer adet Eğitim Seti kazananlar arasında ise *Meral Kösoğlu'*nun ismi yok. Oysa o da bir adet eğitim seti kazandı.
Çekilişle ilgili eksik ve fazla isimlerimiz bu kadar. Bir büyük hata daha ise, Bir Öykümüz Var, bölümünde yayınlanan Otomatik Dev Gramatizör, Amerikalı yazar *Roald Dahl'*ındı. Türkçe'ye *Tomris Uyar* tarafından çevrildi.

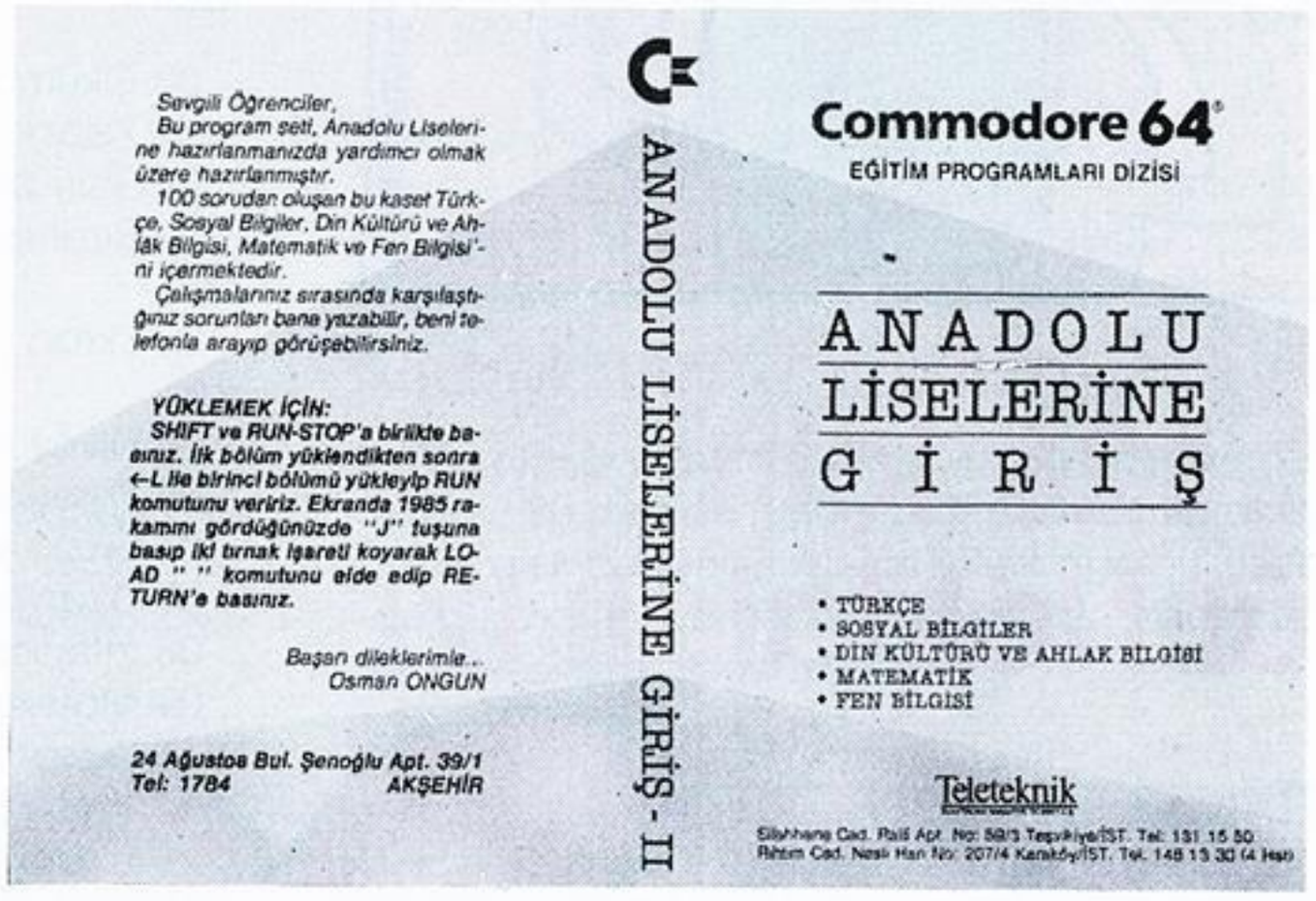
Teleteknik'ten Yeni Bir Eğitim Paketi

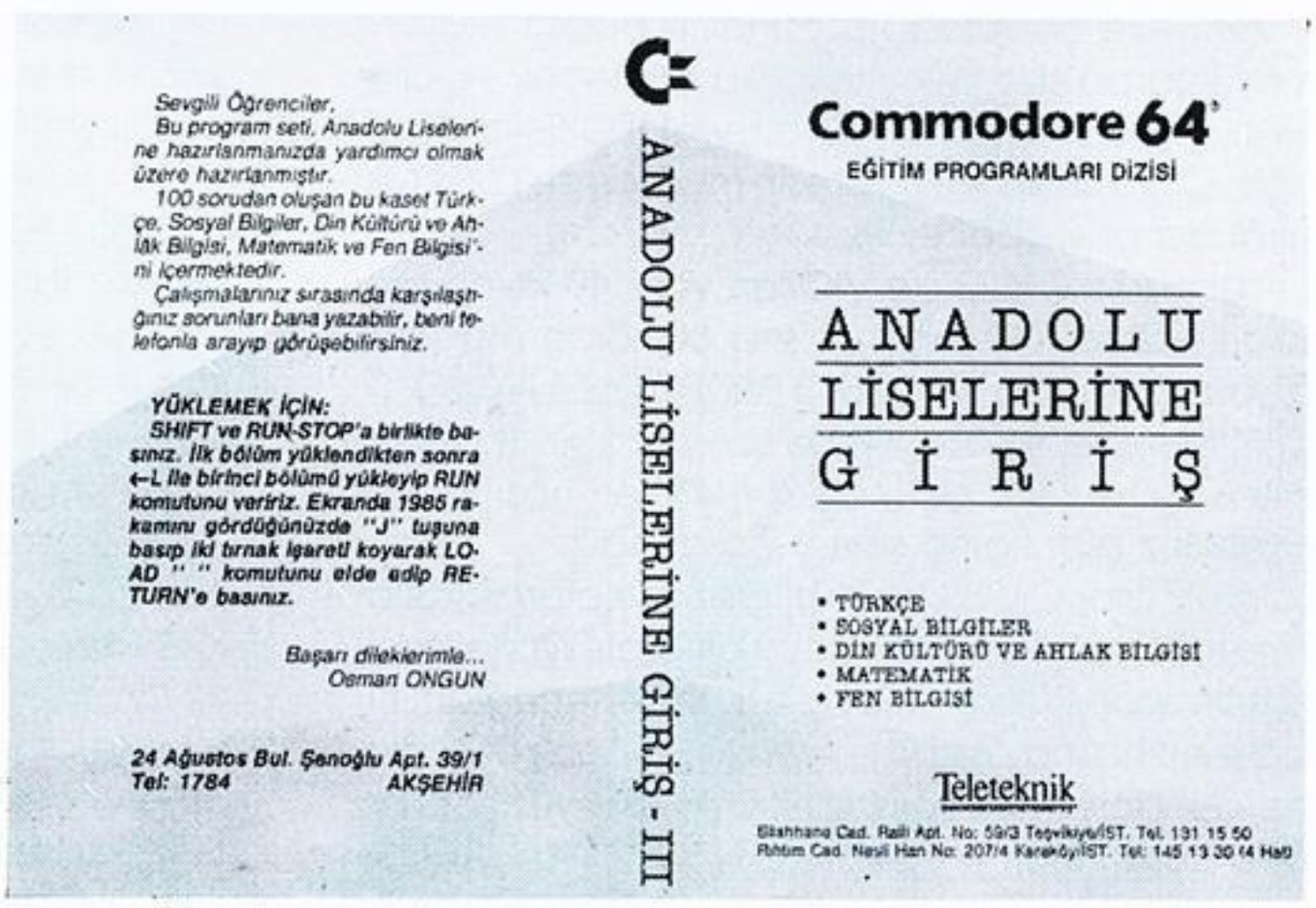
Anadolu Liselerine Giriş

Üç kasetten oluşan bu program seti, Anadolu liselerine hazırlanmanızda yardımcı olmak üzere hazırlanmıştır.

100 sorudan oluşan her kaset Türkçe, Sosyal Bilgiler, Din Kültürü ve Ahlâk Bilgisi, Matematik ve Fen Bilgisi'ni içermektedir.







Bütün Teleteknik Bayilerinde

C 128 için Püf Noktaları (2)

BURAK KİPER

Bu yazımızda C128'in yanında C-64 ile de ilgili bilgiler vermeye çalışacağız, bundan başka ara sıra bazı oyunlarda size kolaylık sağlayabilecek veya size bir oyunda sonsuz hak verdirecek POKE'lar öğreteceğiz. C128 ile ilgili bir adresin mümkünse C-64'teki karşılığı da gösterilecektir.

İlk yazımızda C-128'de VIC'ten yani 40-kolon modundan bahsetmiştik, şimdi sıra 80-kolon moduna geldi, bu yazımızda VDC 8563 (VIDEO DISPLAY CONTROLLER) işlemcisinin özelliklerini tanıyacağız. Bu işlemciye nasıl girebiliriz, normal karakterleri nasıl iki kat büyütebiliriz gibi ilginç şeyler göreceğiz.

Bildiğiniz gibi C128'i bir iş bilgisayarı olarak kullanabilmemiz mümkün, makineye bir profesyonel özelliği kazandıran 80-kolon'luk gösterim yeteneğidir. Bu arada şunu hatırlatalım ki bu işlemci sadece RGB monitörlerde görüntü sağlamaktadır, bu yüzden bu modda görüntü elde edebilmek için C1901 tipinde olduğu gibi RGB çıkışı olan bir monitör kullanmanız gerekmektedir.

Şimdi alışmış olduğumuz ortamın dışına çıkıyoruz, bildiğimiz gibi 40-kolon modda ekran ve renk RAM'i normal RAM de idi ve normal PEEK ve POKE komutunu kullanarak bu işlemciyi istediğimiz şekilde kullanabiliyorduk. Fakat 80-kolon modunda bu rahata sahip değiliz; çünkü 80-kolonun RAM'i normal RAM'de değildir ve normal PEEK ve POKE komutunu kullanarak üzülmenize gerek yok, çünkü size 80-kolonda kullanılan işlemciyi nasıl kullanacağınızı göstereceğiz. Aşağıda işlemcinin yazmaçlarının bir listesi yeralmaktadır.

00 Okuma: Durum, LP, VBlank,-,-,-,-,-
Yazma: istenilen yazmacın 0-5 Biti
01 Satır başına karakter sayısı
02 Ekran penceresini kaydırma (yatay/karakter olarak)
03 Ekran penceresini kaydırma (yatay/piksel olarak)
04 Dikey senkronize
05 Dikey toplam
06 Ekran sayfasında yeralan satır sayısı
07 Ekran penceresini kaydırma (dikey/satır olarak)
08 Interface modu
09 Matriks yazmacı - dikey
10 Kursor modu - taramaya başlama
11 Taramayı durdurma
12 Ekran hafızasının başlama adresi - yüksek byte
13 Ekran hafızasının başlama adresi - alçak byte
14 Kursor'un yeri - yüksek byte
15 Kursor'un yeri - alçak byte
16 Light Pen dikey
17 Light Pen yatay
18 Kanal adresi - yüksek byte
19 Kanal adresi - alçak byte
20 RAM başlangıç adresi - yüksek byte
21 RAM başlangıç adresi - alçak byte
22 Matriks yazmacı/yatay görüntü
23 Matriks dikey görüntü
24 Dikey düzgün akış
25 Yatay düzgün akış
26 Renk
27 Adres kaydırma
28 Karakter üretme basic adresi - yüksek byte
29 Alt çizim - Kursor - tarama - satır
30 Yazmaç tekrarı
31 Kanal, video RAM'indeki byte'ı okuma/yazma

32 Blok başlangıç adresi - yüksek byte
33 Blok başlangıç adresi - alçak byte
34 Ekran gösteriminin başlangıcı
35 Ekran gösteriminin sonu
36 Tazeleme oranı

Şimdi listesini verdiğimiz yazmaçları bireysel olarak örnekler vererek işlevlerini açıklayacağız.

Bu yazmaçlar dolaylı olarak adreslenirler. Bunun anlamı sadece sıfırıncı ve birinci yazmaçlara giriş yapabildiğimizdir. Bunu şu örnekle açıklayalım, eğer 26 numaralı yazmacın içeriğini görmek istiyorsanız şu satırları yazmanız gerekir.

```
A=DEC("D600")
POKE a,26:PRINT PEEK(A+1)
```

0 ($D600) ve 1 ($D601) yazmaçlarına yazılan yazmaç numarası istenilen yazmaçtan okuma ve yazmak için bir kanal açmamızı sağlar. Bunu aşağıda vereceğimiz küçük bir program yapabilir.

```
10 INPUT "YAZMAC";Y
20 INPUT "DEGER";D
30 POKE DEC("D600"),Y
40 POKE DEC("D601") ,D
```

Yazımızın başında da belirttiğimiz gibi VDC işlemcisinin normal adreslenemeyen 16K'lık RAM'i bulunmaktadır. VDC video RAM'ine girebilmek için aşağıdaki programı kullanın.

(Program Döküm Eki/PROGRAM 1)

Satır	Açıklama
100	BA değişkenine VDC'nin temel adresini yüklenir.
110	V değişkenine istediğiniz video RAM adresini yükleme.
120	W değişkenine video RAM'ine yazmak istediğiniz değer yüklenir.
130	V değişkenindeki değer yüksek ve alçak byte'lara ayrılır.
140	18'inci yazmaca istenilen adresin yüksek byte'ı konulur.
150	19'uncu yazmaca alçak byte konur.
160	31'inci yazmaca byte değeri konulur.
170	BA değişkeninin hafıza adresine ulaşana kadar komut bekle.
180	30'uncu yazmaç 1 ile doldurulur (karakter çıkışı).

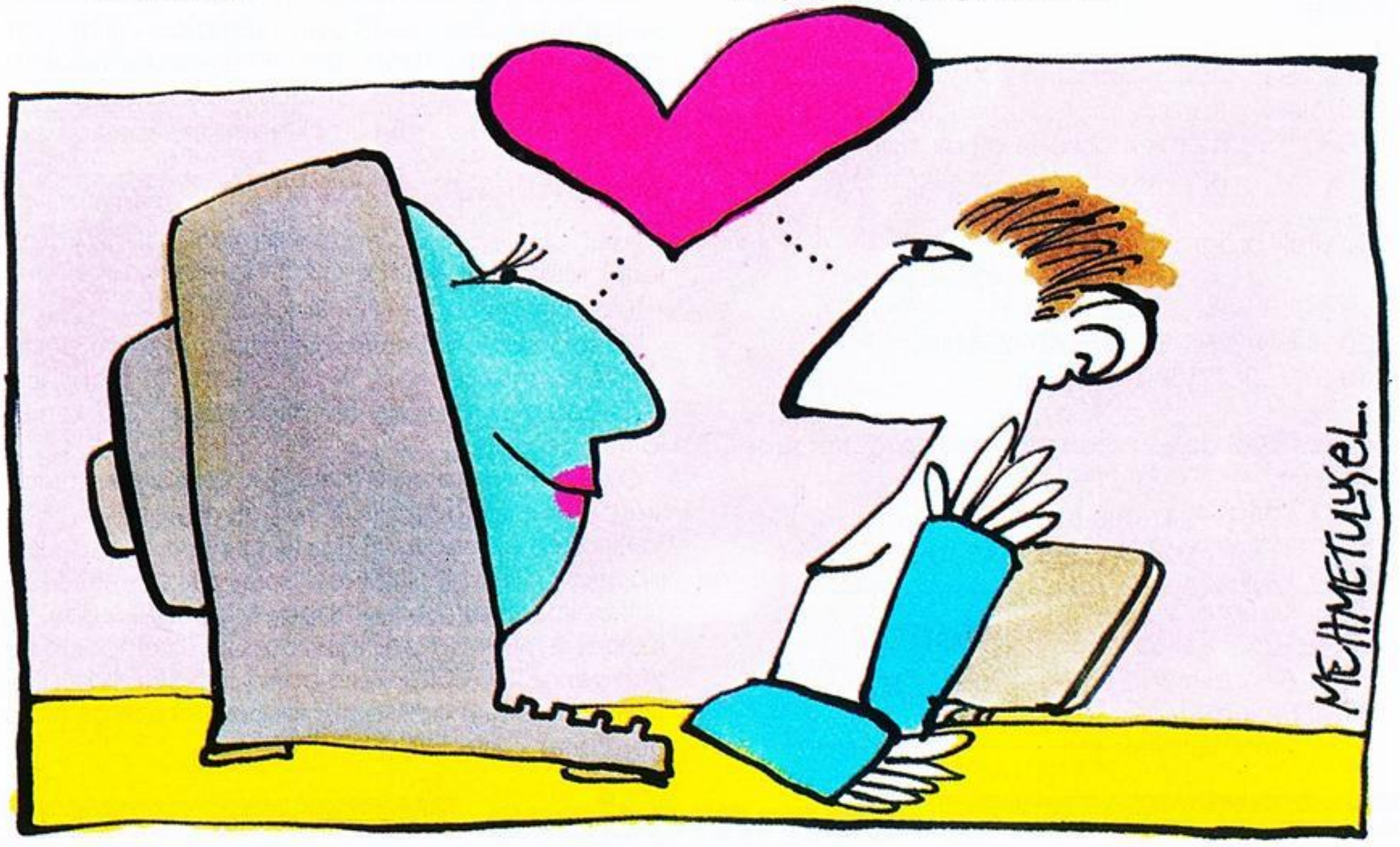
Bilgisayar ilk açıldığında video RAM'in durumu şu şekildedir:

$0000-$07FF Ekran tazeleme hafızası (SCR-RAM) (0-2047)
$0800-$0FFF RAM (mesela ekran hafızası) (2048-4095)
$1000-$1FFF Serbest (4096-8191)
$2000-$3FFF Karakter üretimi (8192-16385)

Yukarıda verilen Basic programı çalıştırın ve adres olarak '0' girin. Değer olarak '0 ve 255' arası bir değer giriniz. 80-kolon gösterimde üst sol köşede verdiğiniz değere bağlı olarak bir karakter yazılacaktır.

Video RAM'e pratik olarak giriş

80-kolonda işletim sistemi bir tuşa basıldığında ne yapar? Bu modda işletim sisteminin bir karakteri ne yaptığını inceleyelim. İşletim sisteminde bu işle ROM'da bulunan bir yordam görevlidir. Bu yordamı BASIC'ten şu şekilde çağırabilirsiniz:



SYS 49155, karakter, renk
Karakter: Karakter numarası (0-255)
Renk : Karakter rengi (0-16)

Bu yordam hem 40-kolon hem de 80-kolon için çalışır, bunun anlamı bir komutla her iki ekranı da aynı anda programlayabilirsiniz demektir. Şimdi karakterleri ekranın çeşitli satırlarına nasıl koyacağımızı öğrenelim. Karakterin yeri hafızadaki 224 ve 225 numaralı (o anki Kursor'un yeri) adreslerden okunabilir. Eğer bir karakterin belirli bir yerde olmasını istiyorsanız, bu adreslerle biraz oynamak yeterli olacaktır.

Aşağıda bunu yapabilecek küçük bir program veriyoruz:

(Program Döküm Eki/PROGRAM 2)

KOL değişkeni: 0-79 (0-39)
SR değişkeni : 0-24

Programda kullanılan bazı adresler şunlardır:

238 Ekranın maksimum uzunluğu
224 Kursor'un yeri - alçak
225 Kursor'un yeri - alçak
49155 Başlangıç adresi - ROM yordamı

Taklit edilmiş POKE

Aşağıda vereceğimiz program sayesinde normalde POKE yapamadığımız video RAM bölgesine bizim yaptığımız POKE komutu ile POKE yapmak mümkün olacaktır.

(Program Döküm Eki/PROGRAM 3)

Hazırlamış olduğumuz POKE komutunun formatı şu şekildedir:

SYS DEC("1800"), karakter, LO,HI
Karakter= Karakter byte değeri (0-255)
LO= İstenilen adresin alçak byte'ı
HI= İstenilen adresin yüksek byte'ı

Şu ufak programı deneyin:

```
10 INPUT"ADRES";AD
20 HI=INT(AD/256):LO=AD-(256*HI)
30 SYS DEC("1800"),3,LO,HI
```

Video RAM'deki herhangi bir byte'ın içeriği şu şekildedir:

BIT 0: Parlaklık
BIT 1: Mavi
BIT 2: Yeşil
BIT 3: Kırmızı
BIT 4: Yanıp sönme
BIT 5: Altı çizilmiş
BIT 6: Reverse video
BIT 7: İkinci karakter seti

İlk dört bit'in kullanımı oldukça açıktır, bu bitleri birleştirerek mümkün olan 16 rengi elde edebilirsiniz. Dördüncü bit'i 1 yaparak karakterinizin yanıp sönmesini sağlayabilirsiniz. Biraz eğlenmek için üstte verdiğimiz programın 30'uncu satırını şöyle değiştirin:

```
30 SYS DEC("1800"),2  0+2  3+2  4,LO,HI
```

Programı çalıştırarak adrese 2048 değerini verin, şimdi ekranda açık kırmızı renkte olarak karakter yanıp sönmeye başlayacaktır. BIT 7 ile karakter setini değiştirebilirsiniz, normalde bunu SHIFT/C= tuşu ile yapıyoruz. 40-kolon ekranın aksine 80-kolon ekranda aynı anda her iki karakter setinin de bulunması mümkündür. Bunun anlamı da çalışabileceğiniz karakter sayısının 512 adet olduğudur.

Karakter üretimi

Aşağıda vereceğimiz program karakter üreticiyi video RAM'den okuyacaktır.

(Program Döküm Eki/PROGRAM 4)

Programımızda kullanılan değişkenler:

BA : Karakter üreticinin başlangıcı
A : VDC'nin başlangıcı
AD: Karakter üreticide o anda bulunan adres
HI : AD değişkenindeki adresin yüksek byte'ı
LD : AD değişkenindeki adresin alçak byte'ı
CH: Karakter satırının okunuş değeri
W : Sayaç

Programı çalıştırdığınızda bütün karakterler ekrana genişletilmiş matrisler halinde basılacaktır. Her karakteri sekiz boşluğun izlediğini dikkat edeceksiniz. Bundan başka 80-kolon karakter üreticinin VIC-20 veya C-64'dünkilerden farklı olduğuna dikkat edin.

Her karakter 18 byte'dan meydana gelmektedir. Bu fazladan olan sekiz byte normalde kullanılmamaktadır, bunlar bazı özel durumlarda kullanılacaktır, VDC 8563 hem 8*8 matrislik hem de 16*8 matrislik karakterler üretebilmektedir.

Büyütülmüş karakterler

Aşağıda yeralan program ile genişletilmiş karakterler elde edeceksiniz.

(Program Döküm Eki / PROGRAM 5)

Bir sonraki yazımızda biraz daha 80-kolon gösterimden bahsedeceğiz, şimdi C-64'te oynayabileceğiniz bazı oyunları kolaylaştıracak yararlı POKE komutlarına geçiyoruz.

Oyunumuzun adı PSYCASTRIA, başlık ekranında iken, yani bilgisayara RUN komutunu verdikten sonra ilk çıkan ekranda I L T D N tuşlarına aynı anda basıp bırakın. Bu tuşlar İngilizce olarak şu kelimelere karşılık gelmektedir: I Love Tracey Dianne Nicolson. Bu tuşlara aynı anda bastığınızda size sorulacak olan şifre yerine THEVDICE kelimesini yazdığınızda CHEATIN MODE'a girmiş olacaksınız, böylece de gemimiz görünmez olacaktır. □

SADECE

commodore

SATARIZ

Garanti Bankası
Bakırköy Şubesi
Hesap No: 1.609.576-2

SORBİM

Bilgi İşlem Merkezi

Teleteknik

commodore

Yetkili Satıcısı

Istanbul Cad. No: 18 D.32
Bakırköy-ISTANBUL
Tel: 583 85 33-583 34 60

MAYIS AYI TEK OYUNLUK KASETLERİMİZ

1) WORLD GAMES
2) TARZAN
3) SILENT SERVICE
4) TWO ON TWO
5) FIST 2
6) ASTERIX
7) INFILTRATOR
8) THE WAY OF THE TIGER
9) THAI BOXING
10) CAULDRON 2
11) LEADER BOARD 3
12) BAZOOKA BILL
13) DAN DARE
14) UCHI MATA
15) 1942
16) STAR PING PONG
17) WAR PLAY
18) PAPER BOY
19) PARALLAX
20) SINBAD
21) HIGH LANDER
22) IT'S A KNOCKOUT
23) ALLEYKAT
24) CAPTAIN KELLY
25) MISSION A.D.
26) STRIKE FORCE COBRA
27) YIE AR KUNG FU 2
28) ACE
29) HYPARABALL
30) INFODROID
31) THE SENTINEL
32) SUMMER GAMES 1
33) SUMMER GAMES 2
34) WINTER' GAMES
35) MOVIE MONSTER
36) SUPER CYCLE
37) JOHNNY REB 2
38) LAW OF THE WEST
39) TOUCHDOWN FOOTBALL
40) KNIGHT GAMES
41) BREAK THRU
42) VERA CRUZ
43) STALLONE COBRA
44) BOBBY BEARING
45) ACE OF ACES
46) REPTON 3
47) GLIDER RIDER
48) DROID
49) MISSION ELEWATOR
50) TRAP
51) S.F.POKER
52) SHORT CIRCUIT
53) W.A.R.
54) GALVAN
55) REBEL PLANET
56) ARCANA
57) SPACE HARRIER
58) ANTIRIAD
59) TIME TRAX
60) SCOOBY DOO
61) EXUINOX
62) GREEN BERRET
63) STRIKE FORCE HARRIER
64) WORLD CUP CAR
65) POWER PLAY
66) SABOTEUR
67) IRIDIS ALPHA
68) SKY RUNNER
69) HEAD COACH
70) KETTLE
71) 10TH FRAME
72) TERRA KRESTA
73) DANTE'S INFERNO
74) FUTURE KNIGHT
75) FOOTBALL OF THE YEAR
76) JUDGE DEREDL
77) LEGEND OF KAGE
78) BOMB JACK II
79) AVENGER
80) THE GRAPHIC AD.CRE.
81) HEARTLAND
82) IC.U.P.S.
83) COMMANDO LİBYA
84) DRACULA
85) JUMP JET
86) NINJA MASTER
87) MIKIE
88) D.T: SUPER TEST (SPOR TEST)
89) AVENGER
90) BISMARCK
91) STAR SOLDIER
92) HE-MAN (ADVENTURE)
93) HE-MAN (ARCADE)
94) TERROR OF THE DEEP
95) HOWARD THE DUCK
96) DOUBLE TAKE
97) THE WIKINES
98) NEMESIS (YENİ)
99) FROST BYTE
100) SHOCKWAY RIDER
101) BIE TROUBLE IN CHINE
102) JAIL BREAK
103) EXPRESS RIDER
104) TOMA HAWK
105) TOP GUN
106) KRAKOUT
107) GREAT ESCAPE
108) RANA RAMA
109) TIGER MACHINE
110) ALIENS
111) MUTANS
112) AMERICAN CUP CHALLANGE
113) SHOLIN'S ROAD

TEK OYUNLUK DİSKETLERİMİZ

1) Destroyer
2) Barbie
3) Gun Ship
4) Silent Service
5) Infiltrator
6) Super Cycle
7) Law Of The West
8) Movie Monster
9) A Sound Odyessy
10) Summer Games I
11) Summer Games II
12) World Games
13) Knight Games
14) Two an Two
15) Gauntlet
16) Aliens
17) Top Gun

İsteyenlere ücretsiz oyun listesi gönderilir

OYUN FİYATLARI

Kaset/Disket tek oyun çekimi.....600
Tek oyunluk hazır kaset.....1200
Tek oyunluk hazır disket.....2500

İSTEK YAPACAKSANIZ

1. İsteklerinizi belirleyiniz
2. Fiyatlarını tespit ediniz.
3. İndiriminizi düşünüz (aboneler için)
4. Tutarını banka hesabına yatırınız.
4. Banka dekontunun fotokopisini ve isteklerinizi bize postalayın
6. İstek mektubunda ad, soyad, adresiniz, varsa telefon numaranızı belirtiniz.
7. Size aynı gün yurt için kargo veya posta ile gönderelim.

BİR YILLIK KURS

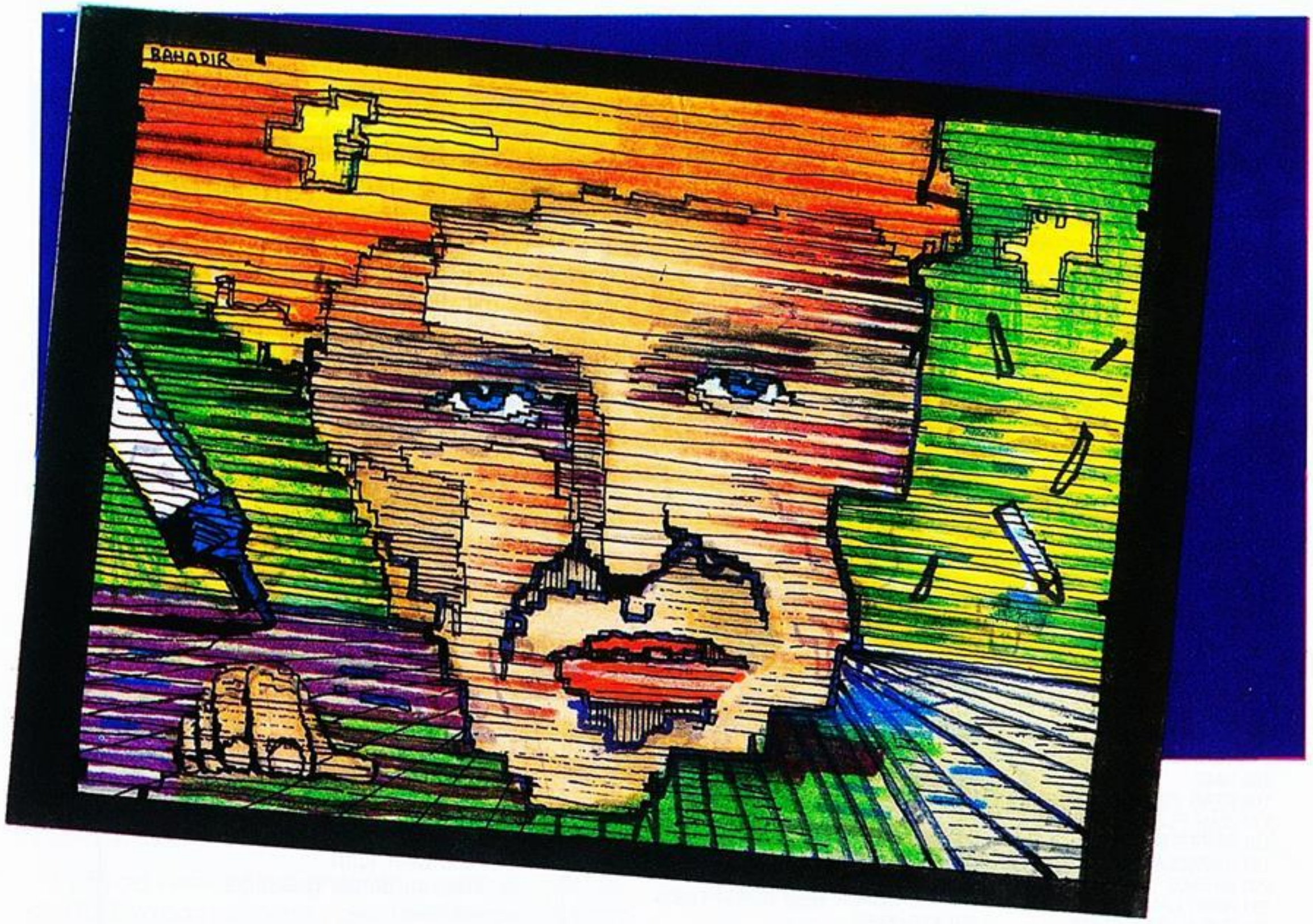
Aylık fasiküller halinde
1. Commodore Basic
2. Commodore Machine Language VE
3. Oyun açıklamaları (her ay 10 ad.)
4. Programlar (her ay 5 ad.)
5. Commodore dünyasından haberler

AYRICA
SORBİM'DE
BİR YIL SÜREYLE
% 10 İNDİRİM

YILLIK 16.000 TL.
POSTA ÜCRETİ ALINMAZ

Abonelere ve Bayilere % 10 indirim

HAZİRAN AYI
REKLAMIMIZI BEKLEYİNİZ



Commodore 64 ile Grafik (7)

TUNA ERTEMALP

Bugüne kadar, altı bölüm boyunca, Commodore 64'ün grafiksel yeteneklerinin başmimarı, görüntü sorumlusu VIC'ın içini dışını, neler yapabildiğini gördük. Tüm yazmaçlar, bellek kontrolü, grafik, sprite derken, sıra onun en son yeteneğine geldi. Bu kez, karakter seti kullanımını, onu renklendirmeyi ve yeni karakterler tanımlamayı öğreneceğiz. Her zamanki gibi, dergimizin 10. ve 12. sayılarında yayınladığımız, grafik dizimizin 2. ve 4. bölümlerini elinizin altında bulundurun.

Bilgisayarınız, sizin her tuşa basışınızda, bastığınız tuşun kodunu, daha önce de gördüğümüz, normalde $0400-$07FF (1024-2047) adresleri arasındaki videoram'a kaydeder. VIC de, ekrandaki görüntüyü yaratırken bu hafıza bölgesindeki bilgiden yararlanır ve sizin devamlı olarak ekranda gördüğünüz görüntüyü oluşturur; hem de saniyede 20 kere. Böylece, görüntünün devamlılığı sağlanır.

Videoram aynı zamanda, eğer noktasal grafik kullanıyorsak, ekrandaki noktaların rengini belirler. Bunu daha önce görmüştük. Bu ayki konumuzla pek fazla ilgisi yok.

Videoram'da saklanan, karakterlerin ASCII kodları değildir. Commodore 64'ün el kitabında da bulabileceğiniz ekran POKE komutlarıdır. Örneğin ekranın sol üst köşesinde A harfi varsa, PEEK (1024) işlemi sonucunda elde edeceğiniz değer 1 olacaktır.

Bu noktada, ekran POKE kodlarını biraz inceleyelim. 0-255 arası ASCII kodlamaya bakarsak, karakterlerin yanısıra, birtakım kontrol ka-

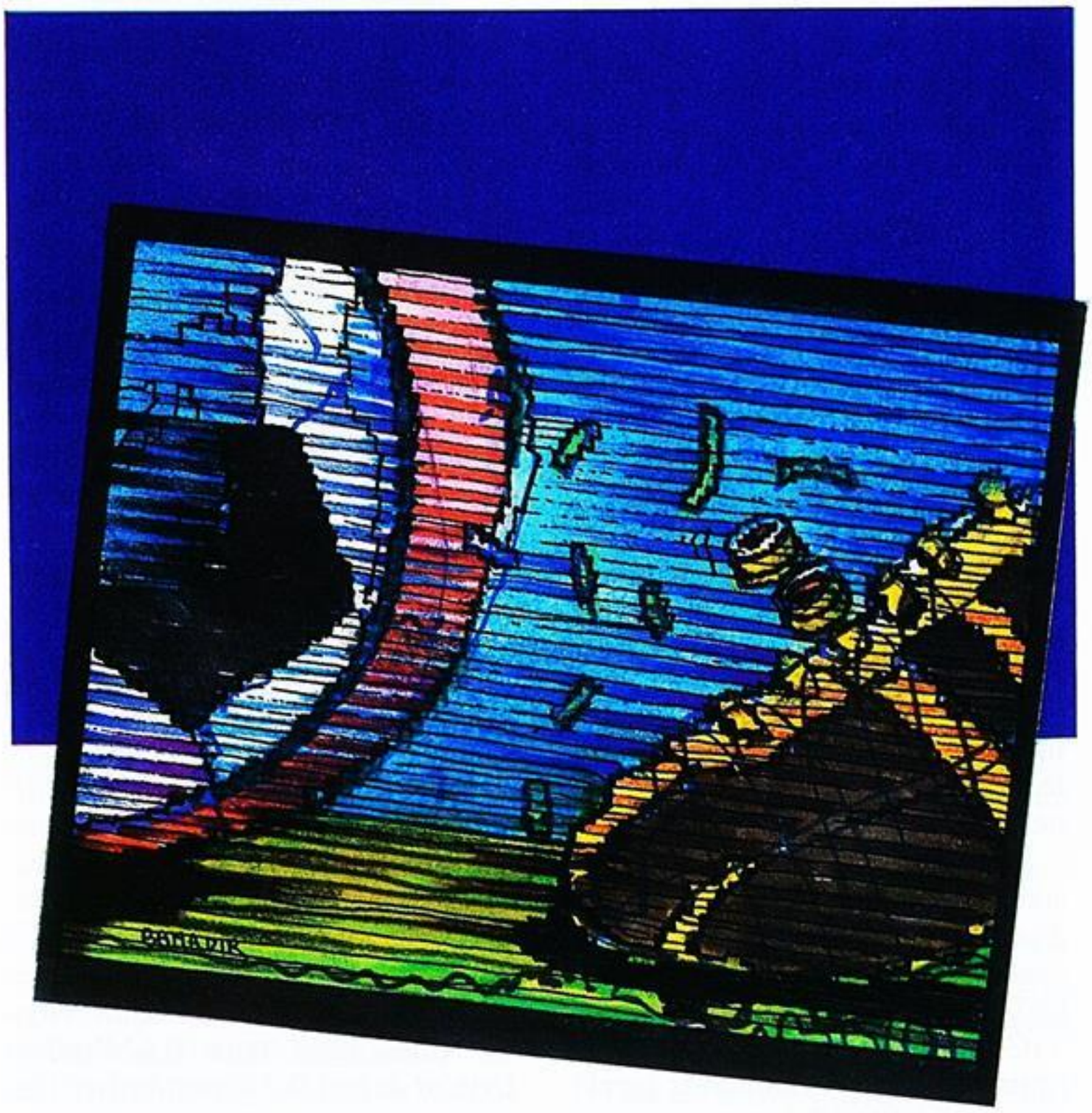
rakterleri de görürüz. Bunlar ekranda hiçbir karakter oluşturmazlar ancak, renk değiştirme, negatif yazıma başlama gibi işlemleri gerçekleştirirler. Ekran POKE kodları ise daha farklıdır. 0-255 arası kodların tamamı karakterlerle doldurulmuştur. Bunun başlıca sebebi, normal karakterlerin yanısıra negatif gösterimdeki karakterlerin de birer koda ihtiyaç duymaları. Böyle bir ayrımın yapılmadığını farzedelim. Bu durumda, VIC, ekrandaki görüntüyü oluştururken videoram'ı okuduğunda elde ettiği bir değerin normal mi, yoksa negatif bir karaktere mi ait olduğunu nasıl anlayacaktı? Bu nedenle 0-255 arası kodların ilk yarısı (0-127) normal karakterlere, ikinci yarısı da (128-255) negatif karakterlere ayrılmıştır. Bu size bir de kolaylık sağlamaktadır. Eğer ekrandaki herhangi bir bölgeyi negatif hale getirmek istersek, videoram'daki o bölgeye ait değerleri PEEK ile okuyup, 128 toplayıp, yeniden yerine POKE etmemiz yeterli olacaktır. PRINT komutunu kullanarak negatif yazmak istediğimizde, RVSON (karakter kodu 18) ve RVSOFF (karakter kodu 146) tuşlarını kullandığınızı hatırlarsınız.

Sıra karakterlerin rengine geldi. Birazdan bahsedeceğimiz renk kodlarının hepsinin 0-15 arası rakamlar olduğunu ve her birinin Commodore 64'ün 16 renginden birine karşılık geldiğini gözönüne alın. Şimdi devam edelim. Renk RAM'ı olarak isimlendirdiğimiz, daha önce de noktasal grafikte çok renklilik kullandığımızda işimize yarayan bu bellek bölümü, $D800-$DFFF (55296-56295) adresleri arasında bulunmaktadır. Bu bölgeyi, BANK adreslemeleriyle oynayarak değiştiremeyeceğimizi geçmiş bölümlerden biliyor olmalısınız. Bu bellek kısmında, ekrandaki her pozisyona bir byte ayrılmıştır (tıpkı videoram gibi). Örneğin 55296'nın içindeki değer ekranın sol üst köşesinin renk kodunu içerirken, 55297 onun hemen sağındaki karakterin, 56295 de ekranın sağ alt köşesinin rengini belirler.

Bu noktada, belki de bugüne kadar dikkatinizi hiç çekmemiş bir konuyu açıklamak istiyoruz. CLR tuşuyla ekranı sildiğiniz zaman Commodore 64, renk RAM'ını baştan aşağı belli bir renk koduyla doldurur. Önemli olan bu renk kodunu nereden aldığıdır. Commodore 64'ün belli bir modeline kadar bu renk kodu, 53281 adresinde bulunan, yazı ekranının rengi olmaktadır. Daha sonra üretilen Commodore 64'lerde ise bu kod, o anda geçerli olan karakter kodunu saklayan 646'dan alınmaya başlandı. Bu değişikliğin ne gibi sonuçları oldu? Eski Commodore 64'te ekranı sildikten sonra 1024-2047 arasına bir değer POKE edildiği zaman bu görünmemekteydi. Çünkü o ekran pozisyonunun karakter rengi, ekran rengiyle aynı olmaktaydı. Tıpkı siyah kalemle siyah kâğıda yazı yazmak gibi. Daha sonra çıkan modellerdeki değişiklik ise şuna yolaçtı: Diyelim ki CLR yapmadan önce kırmızı renkle yazı yazıyordunuz, CLR yaptıktan sonra da ekrana bir şey POKE ettiğinizde, ekranda kırmızı renkle görünecektir. Eski modellerde bu olmayacağı için, CLR'den sonra 55296-56295 arasının da bir renk ile doldurulması gerekmektedir. Tabiî bunun de kolay bir yolu var. Diyelim ki ekranı POKE yaptığınızda karakterlerin sarı olmasını istiyorsunuz. Bu durumda önce POKE 53281,7 ile ekran rengini sarıyapın, sonra ekranı silin. Bu arada renk RAM'ı da sarı rengin koduyla dolacaktır. Sonra da ekranı tekrar eski rengine döndürün. FOR...NEXT döngüsü içinde renk RAM'ının 1000 elemanına POKE yapmaktan çok daha hızlı bir yöntem bu. Eğer elinizdeki model eski ise ve bazı programlar arkadaşınızda görüntü verirken sizde vermiyorsa, programın gerekli yerlerine bu tip bir satır ekleyin. Faydasını göreceksiniz...

Bu minicik (!) ayrıntıdan sonra konumuza devam edelim. Herhalde yazı ekranının renginden VIC'in 33 numaralı yazmaçının (53281) ve çerçeve renginden de 32 numaralı yazmaçın (53280) sorumlu olduğunu belirtmeye gerek yok. Biz daha ciddi konularla ilgilenelim.

Commodore 64'ün içinde iki karakter seti vardır. Birisi büyük harfleri ve grafik karakterlerini kapsar. Diğeri de küçük ve büyük harfleri kapsar ve grafik karakterlerinin bir kısmının yerini harfler alır. Bu iki karakter seti arasında, SHIFT ve COMMODORE tuşlarına aynı anda basarak gidip gelebilirsiniz. Her bir set 256 karakter içerir. Yani hafızada toplam 512 değişik karakter tanımlaması bulunmaktadır. Her bir set de



kendi içinde ikiye ayrılır. İlk yarısı karakterlerin normal görünüşlerini, ikinci yarısı da negatif görünüşlerini taşır. Bu size bir şey hatırlattı mı? Yukarıda bahsettiğimiz ekran kodlarını mesela? Öyleyse, bu setleri biraz daha derinlemesine inceleyelim...

Her bir karakter, 8*8 noktadan oluşur. Commodore 64'ün 320*2000 noktalık bir ekran kapasitesi olduğuna göre, yazı ekranının neden 40'ar karakterlik 25 satırdan oluştuğu ortaya çıkıyor. Peki bu karakterlerinin nasıl görüneceklerini VIC nereden biliyor? Tabiî ki, bilgisayarımızın belleğindeki karakter tanımlamalarından. Her karakterimiz 8*8'lik bir matris oluşturduğuna göre, karakterin görüntüsüyle ilgili bilgiyi 8 byte içinde saklayabiliriz. Toplam 512 karakterinin olduğuna göre 512*8 = 4096 byte, yani 4K hafıza bize yeterli olacaktır. Bilgisayarımızda bu hafıza normal durumda $D000-$DFFF (53248-57343) arasıdır. Karakter tanımlamaları, ekran kodlarına göre sıralanmıştır. Örneğin ilk 8 byte ile "@" karakteri, bir sonraki 8 byte ile de "A" harfi tanımlanır. Bu böylece sürer gider. İlk 256 karakter (karakter kodu 0-265) ile büyük harf ve grafik karakterlerinden oluşan set tanımlanır. Bunun da ilk yarısı normal gösterim, ikinci yarısı negatif gösterime ayrılmıştır. Sonra gelen 256 karakter ise küçük/büyük harf gösteriminin normal ve negatif tanımlamalarından oluşur. VIC ekrandaki görüntüyü yaratırken, videoram'daki ekran kodlarının yardımıyla karakter tanımlamalarındaki doğru yeri bulur ve oradaki karakter tanımını ekrana getirir. Gördüğünüz gibi ekran kodları ile karakter tanımlamalarının düzeni arasındaki benzerlik sadece tesadüf değil. Ancak, karakter tanımlamalarının bulunduğu bu bölge ROM olduğu için üstüne bir şey yazılamaz. ROM'u okuyabilmek için de normal PEEK komutu yeterli olmamaktadır. Birazdan, verdiğiniz programları incelerken bu bölgeyle ilgili zorlukları hazır çözeceğimizi de göreceğiz.

Bilgisayarı açtığınızda, karşınıza çıkan yazı ekranı normal renk modundadır. Ancak, grafik ekranı gibi, yazı ekranını da çok renkli (multicolor) hale sokabiliriz. Bunun için VIC'in 17 ve 22 numaralı yazmaçlarının bit'leri ile oynamamız gerekiyor. Şu iki POKE komutu ile önce

grafik ekranını ve birazdan açıklamasına başlayacağımız extended color (geliştirilmiş renk) modunu kapatıp, multicolor'u aktif hale getiriyoruz.

POKE 53265, PEEK (53265) AND 159

POKE 53270, PEEK (53270) OR 16

Yazı ekranını multicolor'a geçirmenin sağladığı en önemli olanak, özellikle oyun türü programlarda, renkli grafikleri sadece yazı ekranını kullanarak yapabilmek. Grafik ekranının 8K, yazı ekranının 1K olduğunu ve grafik ekranının gerek renk, gerekse nokta organizasyonunun ne kadar karmaşık olduğunu düşünürseniz, multicolor karakterlerin programlamada ne kadar kolaylık yaratacağını kolayca tahmin edebilirsiniz. Şimdi bu özelliğin derinliklerine inelim.

Diyelim ki multicolor'a geçtiniz. Bu durumda bilgisayarınızın kullanabileceğiniz renk sayısı 16'dan 8'e inecektir. Hemen meraklanmayın. Tabiî ki, bunun bize sağlayacağı bir fayda var. Eğer ekrandaki bir karakterin rengi, daha doğrusu onun rengini temsil eden renk RAM'ındaki byte'ın değeri 0-7 arasında ise, sözkonusu karakter normal modda gösterilecektir. Dolayısıyla, multicolor moda geçtiğimiz halde normal karakter biçimlerini kullanma olanağımızı sürdürebiliyoruz. Düşünün, bu olmasaydı, çok renkli yazı ekranında oynanan bir oyunun skorlarını ekrana nasıl yazardık. Sakın multicolor harflerle demeyin! Neden mi? O zaman multicolor karakterlerin açıklamasına girişelim...

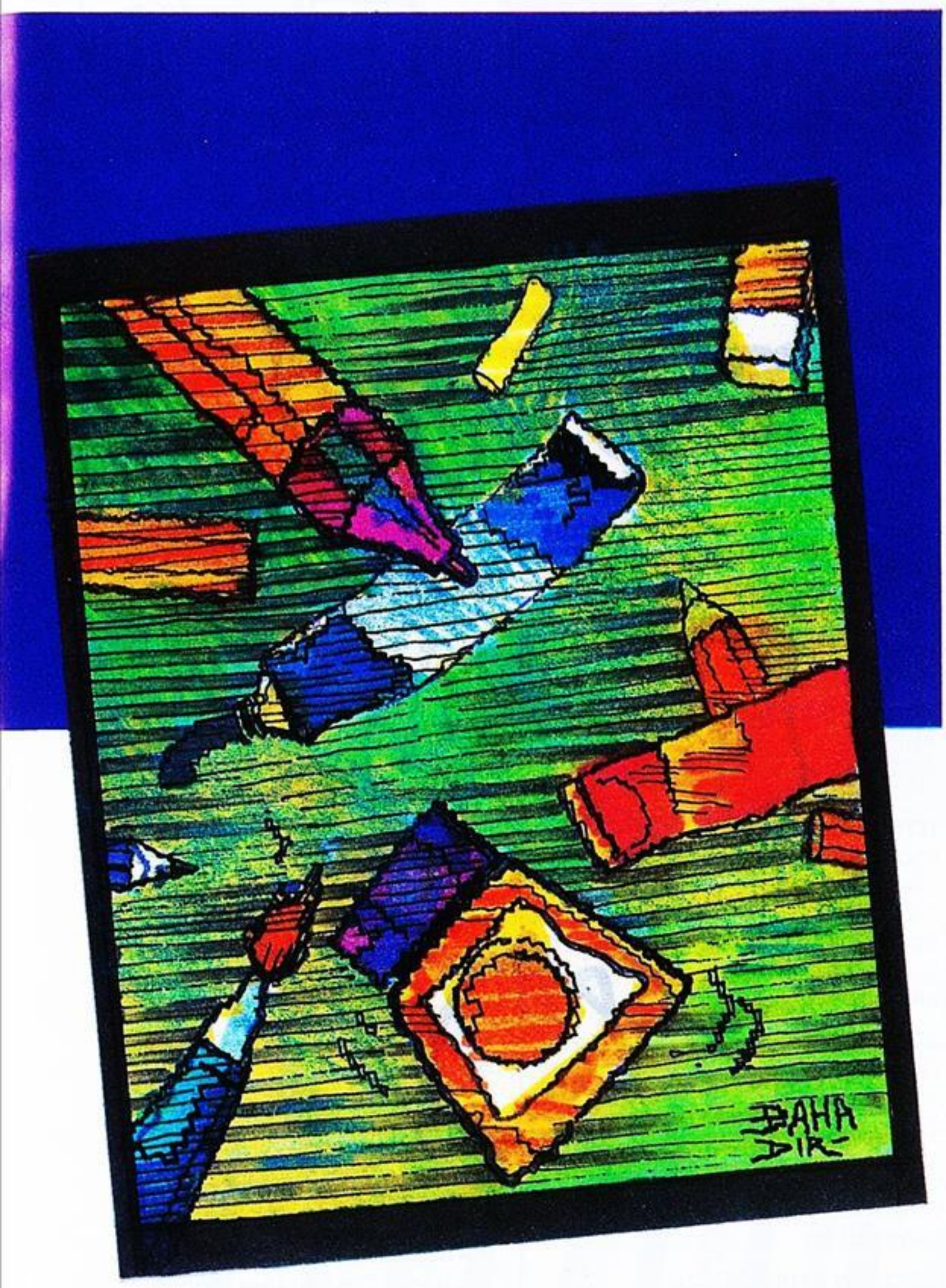
Eğer ekrandaki karakterimizin rengi olarak 8-15 arası bir renk belirlemişsek, artık karakterimiz multicolor'dadır. Yazı dizimizde defalarca belirttiğimiz gibi, multicolor moduna karşılık feda ettiğimiz şey, yatay noktasal çözünürlüktür. Bu durumda karakterimizin çözünürlüğü 4*8'e düşecektir ama, dört değişik renk kullanabilecektir. Her zaman olduğu gibi, yanyana duran iki noktanın aldığı değere göre, oluşturdukları renk noktasının renginin nereden geleceği belli olacaktır. Bir tablo halinde bu ilişkiyi belirleyelim:

%00: VIC'in 33. yazmaçı (0 numaralı arkaplan rengi)

%01: VIC'in 34. yazmaçı (1 numaralı arkaplan rengi)

%10: VIC'in 35. yazmaçı (2 numaralı arkaplan rengi)

%11: 8-15 arası tanımlanan, o ka-



raktere ait renkten 8 çıkartıldığında elde edilen 0-7 arası renk kodu (Renk RAM'ının alt 3 bit'i).

Bu konuyu böylece kapatalım ve oldukça ilginç bir başka özelliğe değinelim: Extended Color.

Bilindiği üzere, ekran renginiz ne ise, ekrandaki karakterlerin arkasında görünen renk de hep odur. Karakteri negatif yapsanız bile, ortaya çıkan şey sadece karakter rengi ile ekran renginin o karakterin kapladığı alan içinde birbirleriyle değiştirilmiş olmalıdır. Peki, her karakterimizin arkasında kalan rengin ayrı olmasını becerebilir miyiz? Tabiî ki... Şüpheniz mi vardı?..

Şu iki POKE komutunu verin:

```
POKE 53265, PEEK (53265) OR 64
POKE 53270, PEEK (53270) AND 239
```

Eğer şansınız varsa, ekranda hemen birkaç değişiklik farkedeceksiniz. Etmediyseniz, şunları sırasıyla yapın:

1) "a" harfine basın.
2) "a" harfine SHIFT ile basın.
3) RVS ON yapın.
4) "a" harfine basın.
5) "e" harfine SHIFT ile basın.
6) RVS OFF yapın.

Göreceğiniz gibi, ekrandaki "a" harflerinin her birinin arkasında farklı bir renk olacak. Birincinin arkasında o anki ekran rengi (normalde koyu mavi), ötekilerinde de muhtemelen beyaz, kırmızı ve camgöbeği olacak.

Ancak dikkatinizi çekti mi acaba? SHIFT ile "a"ya basınca, büyük "A" veya bir grafik işareti çıkmadı. Ortaya çıkan sadece arka rengi değişmiş yeni bir "a" harfi oldu. Şimdi bu özelliği daha teknik açıdan inceleyelim.

Onuncu sayımızda da görebileceğiniz gibi, VIC'in 33-36 arası yazmaçları, değişik arkaplan renkleri tanımlıyorlar.

33. yazmaç: 0 numaralı arkaplan rengi.

34. yazmaç: 1 numaralı arkaplan rengi.

35. yazmaç: 2 numaralı arkaplan rengi.

36. yazmaç: 3 numaralı arkaplan rengi.

0 numaralı arkaplan rengini, normalde ekran rengi olarak kullanıyoruz (53248 + 38 = 53281!). Diğerleri de ya multicolor'da, ya da extended'-de işe yarıyorlar.

Extended moda geçtiğinizde, sadece ekran kodu 0-63 arası olan, karakter setinin ilk 64 karakterini kullanabilirsiniz. Bu tabiî ki büyük bir kısıtlama. Ancak bu setin içine, bütün harf, rakam ve noktalama işaretleri giriyor. Dolayısıyla eğer sadece yazı yazacaksanız, grafik sembolleri ile bir alâkanız yoksa ve arkaplan renklerinin farklı olması önem taşıyorsa, oldukça uygun bir özellik. Bir karakterin alabileceği arkaplan renklerini, ekran kodu'na göre şu şekilde özetleyebiliriz.

0- 63: 0 numaralı arkaplan rengi.

64-127: 1 numaralı arkaplan rengi.

128-191: 2 numaralı arkaplan rengi.

192-255: 3 numaralı arkaplan rengi.

Bir başka açıdan da şunu söyleyebiliriz:

Eğer tuşa doğrudan basılırsa, 0 numaralı arkaplan rengi.

Eğer tuşa SHIFT ile basılırsa 1 numaralı arkaplan rengi.

Eğer RVS ON (CTRL ve 9) ile negatif gösterime geçilir ve tuşa doğrudan basılırsa, 2 numaralı arkaplan rengi.

Yazı ekranıyla VIC arasındaki ilişkiler de bu kadar...

KÜÇÜK İLANLAR

1. kasette Basic (20 ünite) masraf defteri + adres defteri + ilginç programlar + Simon's Basic + SAM Reciter programı

EYLEM CÜLCÜLOĞLU
Tel: 360 30 92.

COMMODORE'nuz var mı? Ya da almayı düşünüyor musunuz? Her türlü sorununuzda bizi arayın.

SECOM BİLGİSAYAR
Cinnah Cad. No: 50/2
Çankaya-Ankara.

COMMODORE satın alınır.
Tel: 525 13 50.
Ucuz Atari
Tel: 525 13 50.

SPECTRUMCULARA müjde!
Spectrumda olmayan Ç, Ğ, İ, Ö, Ş, Ü ve TL. artık ZX48K'da da var. Bu programlar yalnızca 1500.-TL. Ayrıca kasetlere oyun çekilir. 2 oyun 1000.-TL. En son oyunlar liste isteyene verilir.

DEMO SOFTWARE
Tel: 362 75 08.

ACELE satılık Zx Spectrum + (Commodore 64 ve Disk Driver ile takas edilebilir.) 400.000.-TL.

MEHMET ERGÜN ÖKSÜZ
6. Sk. No: 10/3 Pirömer Mh. Ereğli-Konya, Tel: 42320.

C-64 meraklıları için diskette satılık 8 süper oyun, müzik ve orijinal programlar 5000.-TL. Kasette 20 adet çeşitli program ve Simon's Basic 2500.-TL. İngilizce öğrenenler için İngilizce programı (diskette, kasette).

ERDAL KARSLI
Fatih Mh. Hasan Şevket Cad. 51
Sqray Ap. Tel: 91883/1524
Büyükçekmece-İstanbul.

ZX Spectrum Sinclair 250 aşkın oyun + 60 eğitim programı interface + joystick date teybi Commodore ile değiştirmek istiyorum.

TUNA PARFÜMERİ
Bilezikçi Sk. No:72 Pangaltı-İstanbul.

50 Seçme Oyun + Simon's Basic + MDE + BDE + SAM = 5000.-TL (kaset benden). Lütfen liste isteyiniz. Ayrıca isteyenlerle oyun değiştirebilirim. Reset Switch bende 900.-TL'dir.

ERKAN ÜCÜK
İvedik Cad. 9. Sk. No: 12/7 06200 Demet-Gül Ankara, Tel: 345 16 55.

TÜRKÇEYE çevrilmiş III (C-64) ek komutu örnekleri ile Simon's Basic kullanım kılavuzu 3.000.-TL. Simon's Basic kaseti 2000.-Tl. Ayrıca 128 (C-64) ek komutu örnekleri ile Gr-Basic kullanım kılavuzu 4.000.-TL, Gr-Basic kaseti 2.000.-TL satmak istiyorum.

GÜLSÜM TABAK
Akriteks Sentetik İplik San. ve Tic. A.Ş. Kapıkule Yolu 10 Km. Edirne.

TEYBİNİZE çok ucuz fiyata şimdiye kadar eşi görülmemiş bir aygıt takıyorum. Sadece 7500.-TL. Ayrıca 1987 yılının seçme oyunlarını çok ucuz fiyata satıyorum. 1 oyun 500 TL. Telefonlarınızı bekliyorum (Liste isteyiniz). Magic Computer.

TOLGA SAĞLAM
Tel: 140424-144815. Adres: Mithatpaşa Cad. No: 488-490/5 İzmir.

C-64 oyun ve eğitim programları Konya'daki arkadaşlarla program değişebilirim.

ERDOĞAN KAHVECİOĞLU
Tel: 120 58.

FREEZE 1-14000.-TL. Cracker 2 (CR2) 3500.-TL. Supermon (Ex-mon) en son süper monitör programı yalnız 3500.-TL. Ayrıca açık oyunlar Locky J.R., Esra, Turbo 64, Simon's Basic, 64 Mon, Electrosound. Koala Painter 2 vs. çok ucuza.

OZAN
Tel: 1608166

AZ kullanılmış 1541 disk drive. Elektrüsys 64 üniversite hazırlama paketi, Easy Script kelime işlem ve oyun disketleri. Quick shot II joystick (300.000).

Adres: 10280 Altınova Kazan Sitesi Sk. No: 21 Ayvalık/Balıkesir veya Ankara Tel: 392102

ELİMDE bulunan 300'e yakın oyun programını C-64'ü olan arkadaşlarla değiştirmek istiyorum. Elimizdeki oyunlardan bazıları
1) Stallone Cobra
2) Asterix
3) It is a Knockout
4) Future Knight
Liste isteyiniz.

SEROL OLMAZ
P.K. 33 Topkapı. Tel: 34272, İstanbul.

BİR kasette 10 süper oyun + Bateri çalma programı + LOCK J.R. (kilit prg.) + Kopyalama (disk için) + Süper bir org. programı ve spectrum similatör 10.000.-TL. (ödemeli gönderilir)

Adres: Hava Subay Loj. No: 37/3
Tel: 10050, Balıkesir.

BENİM kesinlikle kullanılmamış, içinde 10'dan fazla süper kayıt ve süper yükleme monitör Assembler 64 gibi program olan orijinal Hong Kong malı disket kataloğu ile birlikte 18.000.-TL veya Simon's Basic Kartuş + Kılavuzu ile değişebilirim.

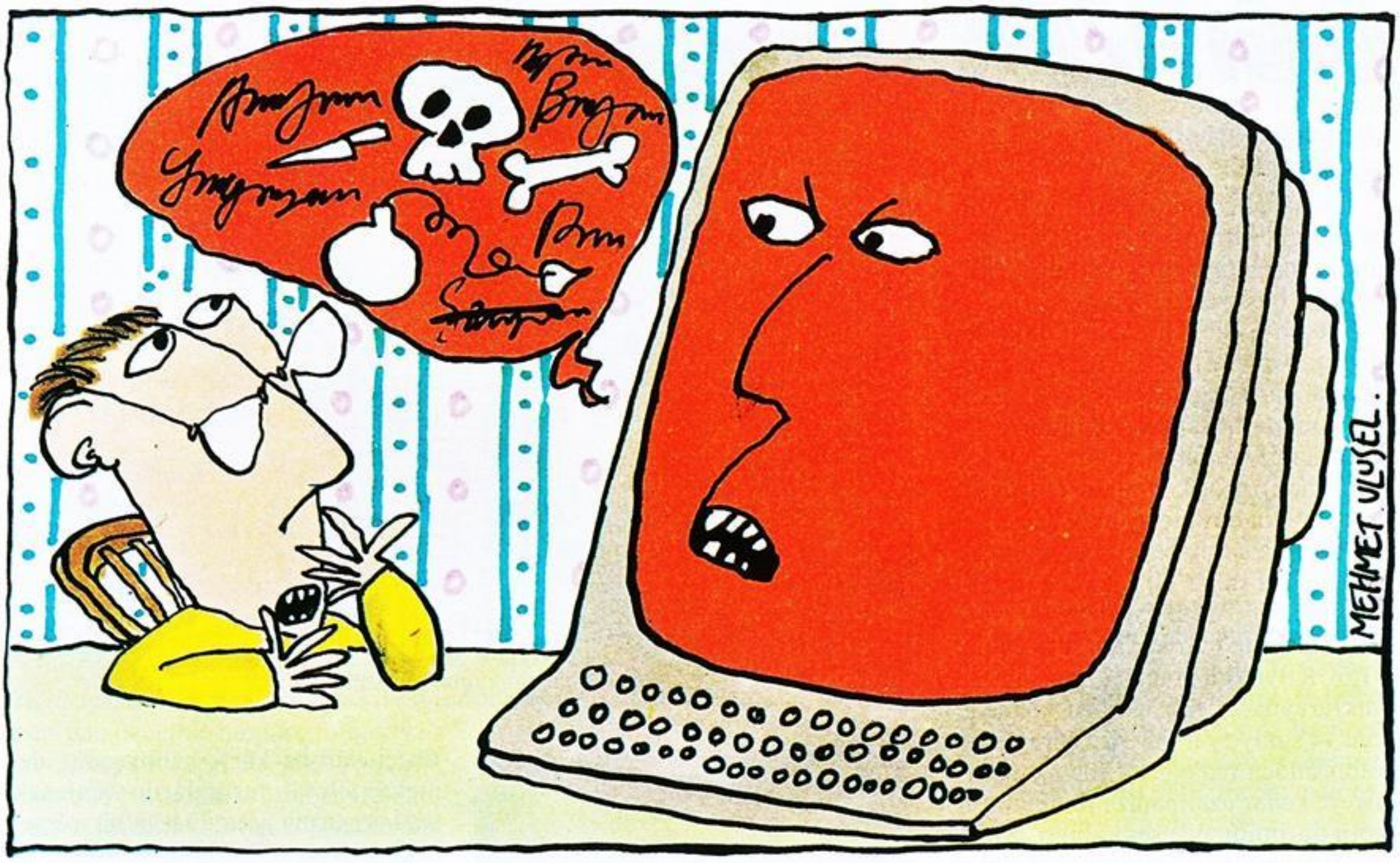
ORHAN COŞKUN
Adres: Hacıilyas Mh. Mecra Sk. Çalışkan Ap., Tel: 4700 Milas/Muğla.

SATILIK C-64 + Datasette Unit + 1 Joystick + CBM 64 User Manual (ing) + 100 oyun + 5 orijinal kasette oyun + Copy 190 + Adres, Masraf, Banka, Ajanda defteri prog. + Simon's Basic + Practical prog. for CBM 64 kitabı + 5 tek oyunlu süper kaset.
Fiyatı: 350.000-TL. veya C-128 ile değiştirilebilir.

AHMET TÜRKER
Tel: 356 52 21, saat 10-17 arası.

SPECTRUM 48K + bilgisayarımı joystick, intirface, kullanım kitabı, İngilizce demo kasedi ile bir C-64 ile değiştirmek istiyorum. Not: Yanında 1 oyun verilecektir.

SERHAT GÜÇ
Eski Londra Asfaltı Uçak Cad. No: 55 Kat: 3 D: 9 Haznedar-İstanbul, Tel: 34600.



SES VE MÜZİK İÇİN DIGITIZER

AYHAN KALAYLIOĞLU

Bu seferki devremiz, çevreden gelen ses bilgilerini dijitale (bilgisayarın anlayacağı biçime) çevirmeyi amaçlıyor. Donanım gereksinimi, benzeri devrelere göre oldukça az olan bu ayki uygulamamızın bir diğer özelliği de oldukça ekonomik olarak gerçekleştirilebilmesi.

Vereceğimiz devre ve kullanım programı (Basic-extension şeklinde) yardımıyla, çevreden gelen birçok ses sinyalini, mesela radyodan, rahat bir biçimde bilgisayarın hafızasına okutmanız sağlanıyor. Hafızanın alabileceği ses bilgisi miktarı, kullanılan örnekleme hızına bağlı olarak değişiyor. Kullandığınız örnekleme hızı, aynı zamanda dijitale çevrilen bilginin ses kalitesini de belirliyor. Yüksek bir hız seçerseniz, örneklemenin kalitesi yükseliyor, fakat hafızayı daha süratli harcıyor. Kullanılabilecek tüm RAM bölgesini bu iş için ayırdığınızda, en yüksek hızda 14, en düşük hızda 45 saniye örnekleme yapmanız mümkün.

Ses örneği bilgisayarın hafızasına alındığında, bu örnek belirli işlemlerden geçirilip işlenip bilgisayarın ses kanallarından verilebiliyor.

RAP, ECHO veya HALL efektlerine benzer efektler elde etmek de mümkün. Bunun için geri verme frekansı ile oynamak veya geri verme hızını değiştirmek veya örneğin geri verilmesi bitmeden tekrar başlatmak gibi teknikler kullanılıyor. Bütün bunları, gerek ufak bir yardımcı BASIC programla, gerek direkt modda elden

vereceğiniz komutlarla yapmanız mümkün.

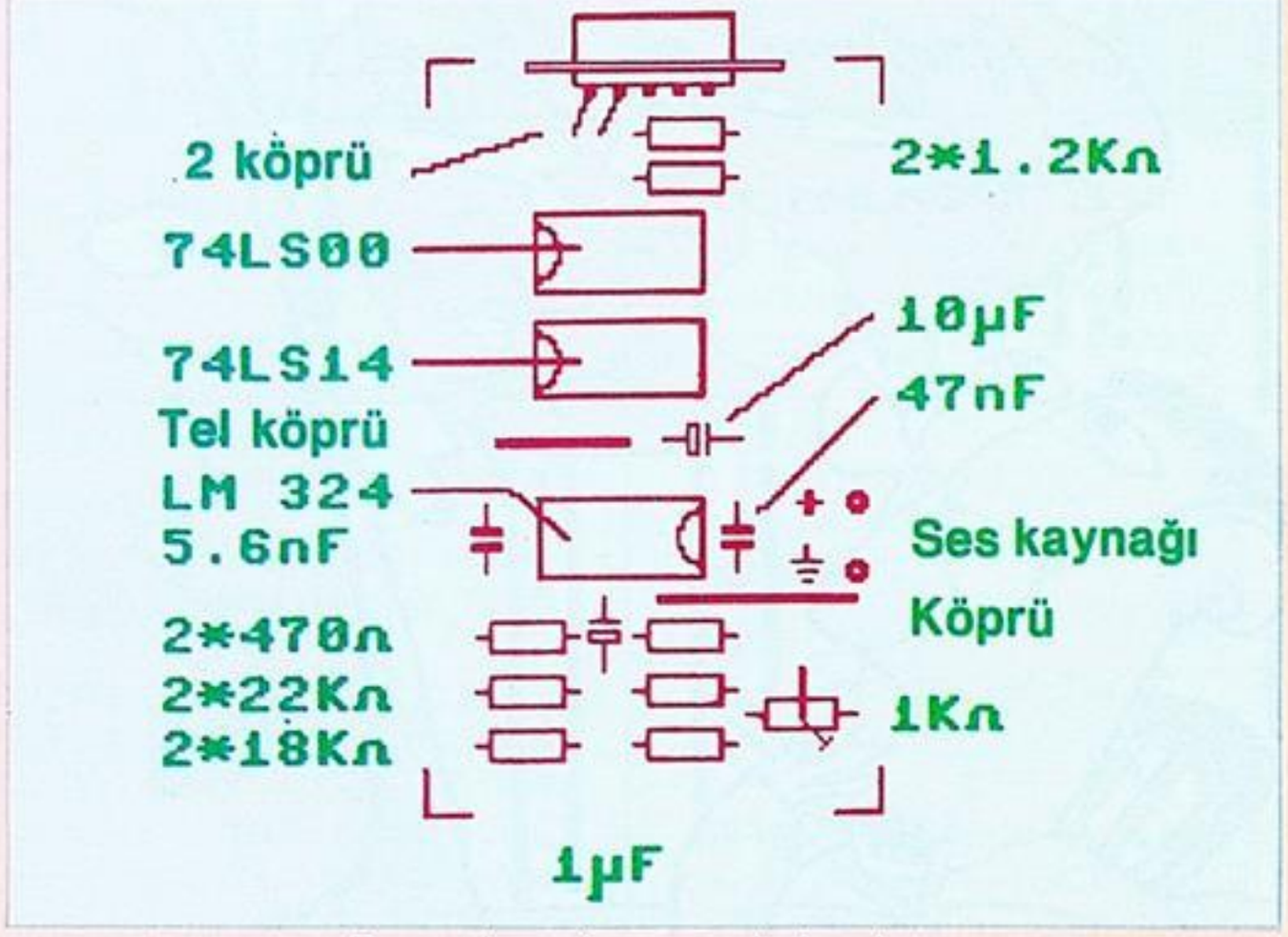
Herhalde çoğunuz, tanıtım kısmını burada bitirip, olayın teknik kısmına geçmemizi istiyorsunuz. Peki öyleyse başlıyoruz!..

Speech-Basic, bir adet iki bitlik ses digitizer devresi ve 4 KByte uzunluğunda Basic-extension (Basic'e özel bir konuda bazı özel komutlar eklemek suretiyle daha zengin bir Basic yaratmak) programından oluşuyor. Her ikisinin birlikte uyumlu çalışması sonucunda gerek sesleri gerek müzik parçalarını rahat bir biçimde örneklemeniz ve işlemeniz sağlanıyor. Speech-Basic için kullanabileceğiniz 57.5 KByte'lık maksimum hafıza miktarı yüksek hızdaki örneklemede bile 14 saniye yetiyor. Bu süreyi, ses kalitesinden fedakârlık ederek 45 saniyeye kadar uzatmanız mümkün. En kötü durumda bile ses kalitesi, SAM ses synthesizer programından daha yüksek!

DEVRENİN GERÇEKLEŞTİRİLMESİ

Devreyi epoksi üzerine, Şekil 1'de verdiğimiz yerleşim planıyla kurarsanız hiçbir sorunla karşılaşmazsınız. Plaketi işlemeye üşenen olursa delikli pertinaksta kullanabilir. Plaketin üzerindeki delikler için 0.8 mm'lik bir matkap ucu en uygunudur. 1 kOhm'luk trimmer için delik çapını biraz daha geniş tutmanızı hatırlatalım. İlk önce rezistanları plakete yerleştirin ve lehimleyin. Böylece sıra kondensatörlere gelir. Tantal tipte olanların kutuplarının doğruluğuna dikkat edin. 10 mikrofarad'lık olanın eksi kutubu plaketin dışına, 1 mikrofarad değerinde olanan eksi kutbu ise LM 324 çipine dönük olmalıdır. Trimmeri de yerine yerleştirdikten sonra JOYSTICK port'una girecek olan fişi monte edin ve lehimleyin. Beş kanallı olan kısmının plaketin yukarısında, dört kanallı kısmının plaketin bakırlı tarafında olmasına dikkat edin. Kuvvetli yapıştırıcılar kullanarak, bu montajı biraz daha da sağlamlaştırabilirsiniz.

Sıra geldi tel köprüleri yerleştirmeye. Bunlardan ikisi JOYSTICK fişinin, plaket ile irtibatını sağlıyor.



Devre elemanlarının yerleşim planı

Üçüncü tel köprü LM 324 ve 74LS14 çipleri arasında yeralıyor. Dördüncü tel köprü, 1 mikrofaradlık kondensatörün eksi kutbunu, girişin şasesiyle bağlıyor. Sinyal girişi ise çift kanallı bir koaksiyel kablo tarafınan yapılıyor. (Tabiî ki koaksiyel kablonun dışındaki zırhı şaseye bağlıyor.)

Herhangi bir stereo ses çıkışını, ses kaynağınız olarak kullanmak için iki seçeneğiniz var. Birincisi yalnızca tek kanalı kullanmak. Bir diğer ise her iki kanalın ses bilgilerini birleştirmek. Pek tabiî ki, bunu stereo çıkışın iki kanalını birbirine kısa devre ederek yapmıyorsunuz. En güvenli yöntem, bu iki ses kaynağının geldiği kanalların üzerine birer 47 nF değerinde kondensatörleri seri olarak eklemek. Böylece, ses kaynağınıza zarar verme olasılığınız ortadan kalkıyor. 47 nF'ların seri olarak yeraldığı yeni ses kaynaklarınızı gönül rahatlığıyla birbirine kısa devre edebilirsiniz. Biraz daha açıklama yapalım isterseniz. Ses kaynağınızın şase çıkışı zaten ortaktır. Onunla ilgilenmeyin. Sizin ilgileneceğiniz, her iki kanalın 'canlı' uçlarıdır. Bunların sonrasına 47'şer nanofarad'lık kondensatörler ekleyeceksiniz. Canlı tarafla hiçbir irtibatınız olmayacak yani. Tekli kullanımlar için zaten devre üzerinde bir izolasyon sağlanmış olduğu için araya hiçbir şey bağlamanıza gerek yok. Diyelim ki ses kaynağınızın şasesini tam olarak bulamadınız, böyle durumlarda, 'köprü çıkışlı' bir ses kuvvetlendirici katıyla karşı karşıyasınız demektir. Böyle durumlarda yapılacak ses karıştırma işlemi başka birçok şeyi de karıştırabileceğinden, sadece ve sadece tek kanalı kullanmanızı salık vereceğiz. Dönelim montaj işinde en son kaldığımız yere. Girişin yapılacağı koaksiyel kablonun montajını yapmıştık en son. Şimdi devre üzerinde tek eksik olan şey çiplerin monte edilmesi işlemi. Burada dikkat edilecek işlem LM 324'ün üzerindeki çentiğin ses girişine, diğer iki çipin de tam ters yöne dönük olması gerektiği.

Gelelim malzemelerin temin işine. Buradaki malzemeler oldukça bol ve rahat olarak, üstelik de ucuza piyasadan temin edilebiliyor. Şimdilik garanti etmemekle birlikte, Teleteknik bu malzemelerin hepsini bir poşet içinde satışa sunacak. Yer olarak (Teleteknik'in Demo-Center binasını kullanacaklar. Poşete baskılı devrenin de dahil olması ve İstanbul dışındakiler için ödemeli gönderilmesi de ihtimaller dahilinde.

Malzeme listemiz aşağıda.

Yarı iletkenler:

1 LM 324
1 74LS14
1 74LS00

Kondensatörler:

1 5.6 nF
1 47 nF
1 1 mikrofarad/16V Tantal
1 10 mikrofarad?16V Tantal

Rezistanslar:
(Hepsi 1/4 Watt, %5 tolerans)
2 470 Ohm
2 1.2 kOhm
2 18 kOhm
2 22 kOhm

Trimpotlar:
1 1 kOhm (yatık pozisyonda)

Diğerleri:
1 Plaket
3 Entegre devre soketi (14 bacaklı)
1 JOYSTICK fişi (9 kanallı)
1 Koaksiyel kablo
1 Ses kaynağına bağlamak için fiş (kullanacağınız ses kaynağına özel ve mono)

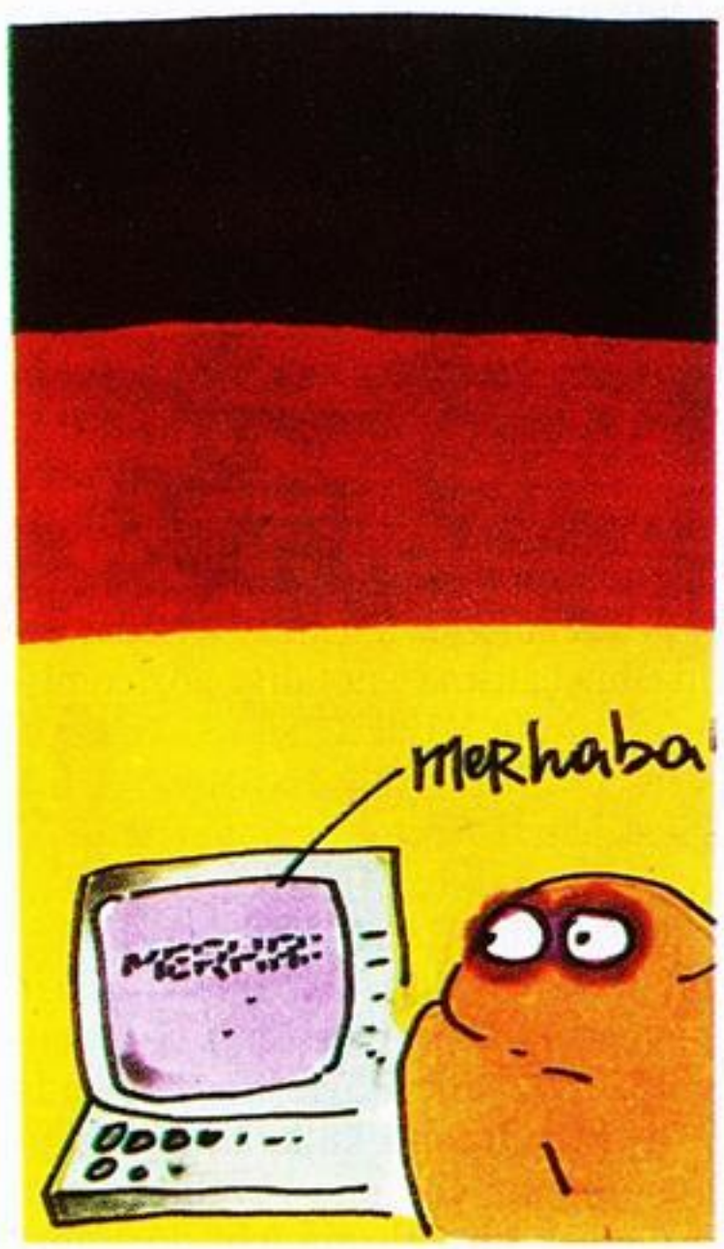
ÇALIŞMA PRENSİBİ

Algıladığımız ses, havada oluşan 20 ila 15000 Hertz frekanslı dalgalardır. Bilgisayarın bu dalgaları algılayabilmesi için özellikle bunları bilgisayarın algılayabileceği bir şekle sokmak lazımdır. Havadaki ses dalgalarını algılayan ilk birim bir mikrofondur. Bu ünite, algıladığı hava basıç değişimlerini (sesin başka bir tanımı) çok zayıf elektrik sinyallerine dönüştürür. Digitizer denilen ünitenin görevi ise kesiksiz ve seviyesiz olarak gelen elektrik sinyallerini seviyeli olarak dijitale çevirmektir. Dijital ile analog arasındaki farkı göstermek için Şekil 3'ü yayınladık. Digitizer'ın çözünürlüğü ne kadar yüksekse sinyal o kadar çok seviyeyle temsil edilir. Yani, daha açık olarak belirtirsek ses kalitesi o kadar yüksektir. Tabiî bu durumda kullanılacak hafıza miktarı da artacaktır. Bizim devremiz iki bitlik çözünürlüğüyle dört seviyeyi temsil edebiliyor. Bu sayının, bilgisayarın hafıza kapasitesinin kullanımı ile, kulağın istediği ses kalitesi arasındaki dengeyi kurduğu inancındayız.

Digitizer devremizi, bir walkman'ın, radyo-teyp sisteminin veya müzik setinin kulaklık çıkışına bağlayabilirsiniz. Devreye ulaşan ses sinyali üç adet işlemsel kuvvetlendirici ile dört değişik seviyeye ayrılır ve bu seviyelerle karşılaştırılır. Seviye karşılaştırmalarının temiz birer dijital seviyeyi temsil etmeleri için her işlemsel kuvvetlendiricinin çıkışına birer ters çeviren tetikleyici kapı konulmuştur. Geriye kalan son çipin içindeki mantıksal kapıların görevi, karşılaştırmalar sonunda elde edilen üç adet sinyali iki bitlik dijital bilgi olarak toparlıyor. Bu şekilde dört değeri temsil edebiliyoruz. Aşağıda sırasıyla, her değeri gösteren iki bitlik çıkışları ve karşılaştırmada elde edilen ara değerleri bulacaksınız.

1. değerde bitler:00; karşılaştırma:111
2. değerde bitler:01; karşılaştırma:110
3. değerde bitler:10; karşılaştırma:100
4. değerde bitler:11; karşılaştırma:000

Devremiz oluşturulan bu iki biti 2 numaralı JOYSTICK port'unun UP (yukarı) ve DOWN (aşağı) kanallarına gönderiyor. Buradan da bu iki bit saniyede 18000 kereye kadar okunabiliyor. Gelen bilgilerin dördü biraraya getirilip bir Byte olarak paketlenip hafızada saklanıyor. Sesin daha sonra geri verilme işlemi SID çipinin $D418 (=54272=54248+24) adresli ses şiddeti ayarlama register'i üzerinden yapılıyor. Söz konusu re-



gister'in bir özelliği, herhangi bir ses kanalı açık olmasa da, buranın değeriyle oynayarak bu register'i dijitalden analoga çeviren bir ünite olarak kullanabileceğimiz. Ses geri verilirken sadece bilgisayarın ses kanalları kullanıldığından devrenin takılı olmasına gerek yoktur. Yani siz bir sesi kendi sisteminizle örnekleyip başkasına bir program olarak gönderebilirsiniz. O da, digitizer devresi olmamasına rağmen sesi dinleyebilir. Programın sağladığı bir avantaj, elde edilen değişik örnekleme seviyesine karşılık gelecek ses şiddeti değerini (0'dan 15'e kadar) sizin belirlemeniz. En basit kombinasyon tipleri olarak 6 değişik lineer kombinasyonu gösterebiliriz. (Hepsinin O olması da dahil, bu durumda geri verilen ses duyulmaz.) Bu lineer kombinasyonlar:

0, 0, 0, 0
0, 1, 2, 3
0, 2, 4, 8
0, 3, 6, 9
0, 4, 8, 12
0, 5, 10, 15

olarak gösterilebilir. İsterseniz daha değişik ve lineer olmayan kombinasyonlar da deneyebilirsiniz. Elde edeceğiniz neticeleri beğenmenize bağlı sadece. İsterseniz bu kombinasyonları dinamik (ses geri verilmekte iken) olarak da değiştirebilirsiniz. Böylece demin yukarıda sözünü ettiğimiz HALL ve ECHO efektlerini elde etmeniz daha da kolaylaşır.

İLK SEFER

İlk kaydımızı yapmadan önce donanımımızın bir fonksiyon testi ve basit bir ayar yapmak gerekli. Bunun için önce "Speech-Basic" adlı listing'i MDE ile yazın ve diskette saklayın. Ayar işlemini yapmanız için bir de ufak ayar tornavidasına ihtiyacınız olacak.

Bilgisayarınızı kapatıp, digitizer devrenizi JOYSTICK-PORT2'ye takın. Henüz herhangi bir ses kaynağı bağlamayın. Bilgisayarınızı açıp tüm tuşları kontrol edin. Tüm tuşlar gayet güzel çalışıyorsa sorun yok! Böylece devrenin yapımında büyük bir hata yapmadığınızı garanti ettiniz. Şimdi Speech-Basic programını;

LOAD"SPEECH-BASIC",8

ile yükleyin ve RUN komutunu kullanarak çalıştırın. Bilgisayar, hemen "C 64 SPEECH SYSTEM V2.7" mesajı il cevap verecektir. Speech-Basic mesajlarını daimi küçük harf modunda verir. Diyelim ki Speech-Basic herhangi bir şekilde bilgisayarın kontrolunu kaybetti (mesela [RUN-STOP+RESTORE]'den sonra). Böyle durumlarda, yeni bir komut olan RESET komutunu kullanarak Speech-Basic'in birçok şeyi düzenlemesini sağlarsınız. Bu arada ha-

DIGITIZER

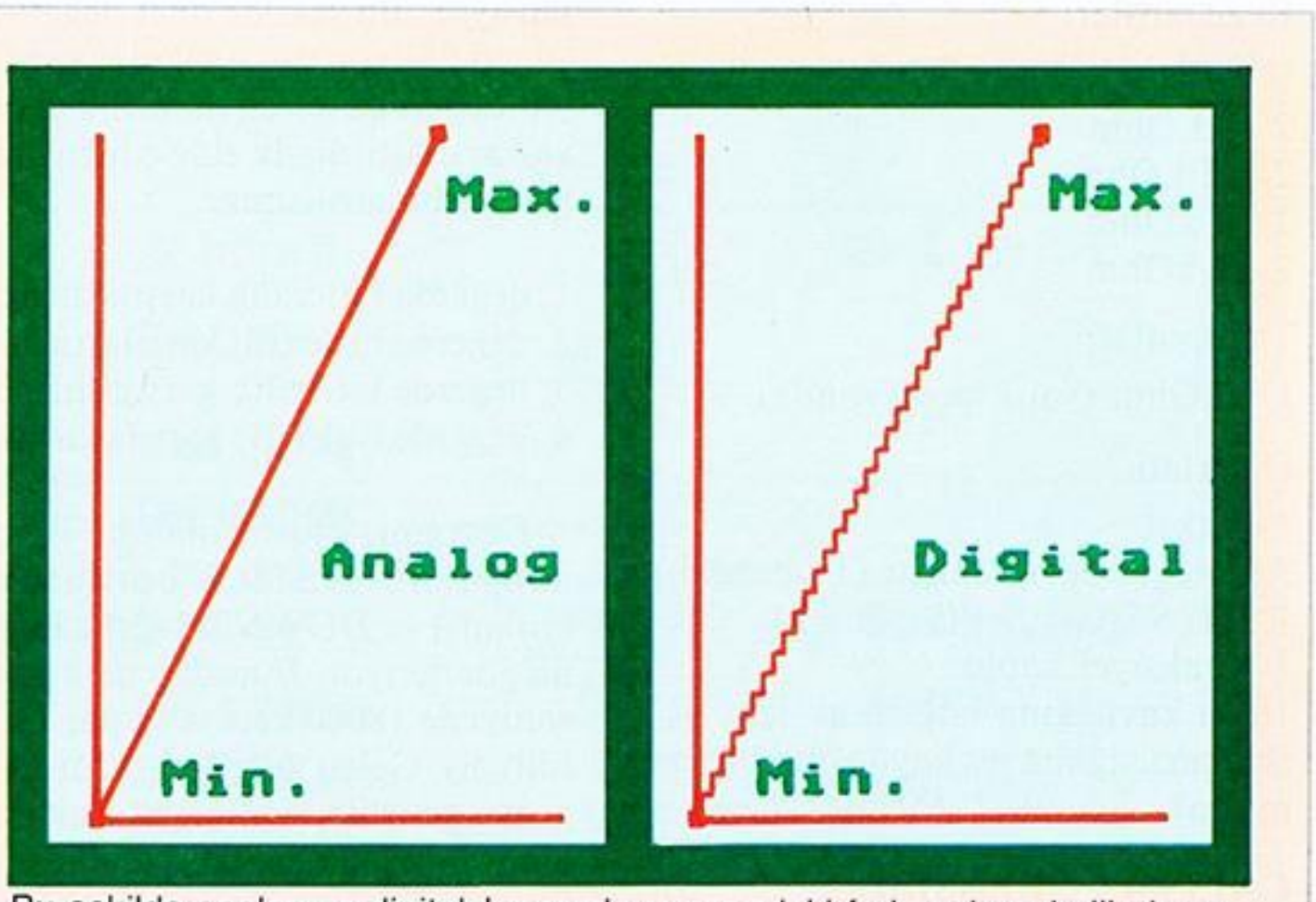
fızada bulunan program silinmez. Bazı Commodore 64'lerin RESTORE tuşu içiçe birkaç tane NMI yaratıp programın kafasını karıştırabilir. Böyle durumlarda da RESET komutunu kullanmanızı tavsiye ederiz. Çünkü bu komut hiç olmazsa bilgisayarı belli bir duruma getiriyor.

Örnekleme seviyelerini renklerle de belirleyebiliriz. Bunun için örnek olarak COLDEF0,11,12,15 komutunu vererek dört değişik gri tonunu istediğimiz renk tonları olarak belirleriz. Şimdi HEAR SPEED 1 komutunu verdiğinizde, girilen sesi dinleme durumuna geçersiniz. Bu işlem sırasında da ekran kapanır (kasetten yükleme yapılır gibi) ve tanımlanmış dört renkten birini gösterir. Şimdi trimpotu en soldaki pozisyonuna getirip, yavaşça en soldaki pozisyonuna doğru hareket ettirirseniz, demin tanımladığımız dört renk aynı sırayla ekranda görünür. Şayet gerçekten bu durum gerçekleşirse devreniz kusursuz çalışıyor demektir.

Tam kalibrasyonu yapmak için trimpotu bu sefer oldukça yavaş olmak üzere tekrar sola çevirin. Kalibrasyonu gözle izleyin ve 3. renkten, 2. renge (bu örnekte griden koyu griye) geçtiği anda bırakın. Bu nokta sizin kalibrasyon (sıfır) noktanızdır. Başka bir nokta da kalibrasyonu çok abartmayın. Mesela çok uğraşıp bu iki renk arasındaki çok hassas bir geçiş bölgesi bulabilirsiniz. Bu geçiş bölgesi o kadar hassastır ki, digitizer'ın girişimde hiçbir sinyal olmasa da ekranda her iki renk gidip gelir. Böyle bir durum aşırı hassaslıktan başka bir şey değildir ve her şeyin aşırısı gibi o da zararlıdır. Böyle bir ayarla yapacağınız örnekleme sırasında bol bol gürültü elde edersiniz, bilhassa ortamda hiçbir şey yokken. Girişi dinleme moduna bitirmek için boşluk tuşuna basın. Böylece bilgisayar normale döner.

SESİN ÖRNEKLENMESİ

Speech-Basic'in 4 KByte uzunluğunda bir program olduğunu söylemiştik. Bu program kendi başına hafızanın $0800'den $1800 adresine kadar kaplar. Bunu hemen arkasından Basic program ve değişkenleri için kullanılan hafıza gelir. Bu hafıza böl-



Bu şekilde analog ve digital kavramları arasındaki fark açıkça belli oluyor.

gesinin bir sonu vardır ve işletim sisteminde HIMEM diye adlandırılır. Normal olarak bu sınır $A000 (40960) adresinde bulunur. İsterseniz bu sınırı aynı isme sahip (HIMEM) komut ile istediğiniz bölgeye (mantıklı olmak koşuluyla) oturtabilirsiniz. Speech-Basic, ses bilgilerini HIHEM değerinden başlayıp hafızanın (aşağı, yukarı) son adresine kadar saklıyor. Böylece HIHEM değerini değiştirmekle, aynı zamanda hafızanın BASIC ve ses bilgileri arasında paylaşım oranını da belirliyorsunuz. Direkt modda yapılan çalışmalarda ne kadar çok ses hafızası ayrılırsa o kadar rahat bir çalışma yapıldığı gözlenmiştir. Örneğin HIHEM $1A00 aşağı yukarı 57.5 KByte'lık hafızayı ses bilgileri için rezerve ediyor. Ayrıca BASIC'te yazabileceğiniz kontrol programı için henüz 512 Byte'lık yeriniz mevcut. Diyelim ki o anlık hafıza paylaşımını bilmiyorsunuz ve bu bilgi size gerekli. Böyle bir durumda MEM komutunu kullanarak sözkonusu bilgiyi ekranda görebilirsiniz.

Gerçek bir çalışma için dört adet örnekleme değerlerine karşılık gelen, gerek ses değerleri, gerekse renk değerlerini belirlemeniz gerekir. Örnek olarak başlangıçta COLDEF0,11,12,15 ve VOLDEF0,5,10,15 komutlarını kullanın. Tecrübe ile daha değişik kombinasyonlar bulursanız bunları da kullanabilirsiniz. Diyelim ki WALKMAN ile çalışıyorsunuz. Devre ile ses kaynağı (burada walkman) arasındaki bağlantıyı kurun ve kaseti istediğiniz bir yere sarın. Walkman'in üzerinden sesi ve bas sesleri sıfıra indirin (sonuna kadar kısın). Şimdi;

HEAR:RECORD SPEED 1

veya F4 tuşu, verin. Bilgisayar gene dinleme durumuna geçecektir. Fakat bu sefer işlem daha ciddi olarak gerçekleşecektir. Şimdi teybi (walkman'i) başlatın ve ses şiddetini yavaş yavaş açın. Bu arada renk bantlarının ekran üzerindeki dağılımına dikkat edin. Gözlemler sonucu elde edilen tecrübeye göre en iyi sonuç tüm renklere ait bantların ekranda aşağı yukarı aynı sıklıkta bulunduğu ses şiddeti değeri sırasında elde edilmektedir. Bu arada bas seslerin şiddeti ile oynayın. Bu arada dikkat çeken bir olay bas seslerin belli bir değerin üstüne çıktıklarında tüm sesi örttükleri. Tüm bunları gözönünde bulundurarak bu parça ile ilgili ses ve bas şiddet ayarlarını optimal sonucu elde edene kadar yapın. Optimal ayarı elde ettiğiniz anda bandı tekrar aynı yere sarıp bir daha başlatın. Şimdi boşluk tuşuna bastığınızda örneklenen sesler hafızaya kaydedilmeye başlanacaktır. Aşağı yukarı 14 saniye sonra hafıza dolacak ve bilgisayar tekrar ekranını açıp mesaj verecektir. Hafızaya okuma işlemini, ara bir yerde durdurmak istediğinizde ise yapacağınız sadece (RESTORE) tuşuna ((RUN-STOP)'suz) basmak. Böylece direkt moda geri döner ve hafızaya okuma işleminin kesildiği adresi belirten mesajı alırsınız.

Eğer yapacağınız kaydın tüm hafızayı doldurmasını istemiyorsanız, o

zaman RECORD komutundan sonra kayıt sırasında doldurması için bir adres aralığı vermeniz gerekiyor. Bunun kullanımı ise şöyle:

RECORD FROM başl.adr.
TO bit.adr.SPEED x

Burada başlangıç adresi ve bitiş adresi HIMEM değerinden büyük olmalı ve tabiî ki bitiş, başlangıçtan ileride olmalı. En son parametre x ise 0'dan büyük olması dışında sınırlamaya tâbi değil. Örneklenmiş sesin geri verilmesi PLAY (ya da F3 tuşu) komutu ile yapılıyor. Buna ait parametrelerin kullanımı ise RECORD komutundakilerin aynı. Detaylara daha ileride değineceğiz.

DİSKET İŞLEMLERİ

Demin de belirttiğimiz gibi, Speech-Basic her iki ROM'un altındaki RAM'i oldukça yoğun bir biçimde kullanıyor. Ortaya ufak bir sorun çıkıyor tabiî ki. İşletim sistemindeki standart LOAD ve SAVE rutinleri bu bölgeler için kullanılamıyor. Bu yüzden sözkonusu rutinler bir kere daha yazılmış durumda. Yeni komutlarımızın adları BLOAD ve BSAVE ve sadece RAM'e hitap ediyorlar. Her iki emir sadece disket işlemleri için kullanılıyor. Zaten başka bir türlüsü de saçma olurdu! Düşünün bir kere, 60 KByte'ı tek parça olarak kasete saklamışsınız. Ses bilgileri için ayrılmış hafıza bölgesini tümüyle diskete saklamak için sadece 'BSAVE"isim"' demeniz yeterli. Bu komutu tüm parametreleri ile görmek istersek:

BSAVE"isim",sürücü no.su
FROM başl.adr. TO bit.adr.

En sondan başlayarak, başa doğru tüm parametreler istenilse verilmeyebilir. Böyle bir durumda o parametre için bilinen belli bir değer kullanılır.

BLOAD komutunun parametreleri ile birlikte kullanımı BSAVE komutu ile aynı. Burada kullanılan adres, disketten alınan bilginin hangi adreslerin arasına gideceğini belirtir. Bitiş adresi bir sınırı temsil eder ve yükleme sırasında hiçbir türlü aşılmaz. Mesela, yüklenen dosya, belirtilen aralıktan daha büyükse, dosyanın geri kalan kısmı dikkate alınmaz ve yükleme işlemi orada kesilir. Böylece, yükleme işlemi üzerinde de kesin bir kontrol elde edilir.

Bunlardan başka, Speech-Basic her türlü disket işlemini rahat olarak gerçekleştirebilmeniz için DISK ve DIR gibi komutlar da sunmakta. Bunlardan ilki, disket sürücüsüne emir göndermek için, diğeri de disketin içeriğini görmek için kullanılıyor.

SINIRLANMIŞ ÖRNEKLER, BLOK TABLOSU

Speech-Basic'te her bir kelimeyi,

cümleyi veya istediğiniz ses bölgelerini bulup, izole etmek oldukça kolaydır. Bunun için, örneklenmiş sesi iki kere dinleyip, birinde kelimenin başında, diğerinde ise sonunda (RESTORE) tuşlarına basıp, başlangıç ve sonuç adreslerini belirlemek gerekiyor. ((RESTORE)'a basıldığında dinleme işlemi kesiliyor ve kesilme adresi belirtiliyordu.) Kelimenin bulunduğu bölgeyi belirledikten sonra, bu bölgeyi,
FROM başl.adr. TO bit.adr.
komutunu vererek dinleyebilirsiniz. Burada, başlangıç ve bitiş adreslerini biraz değiştirerek birkaç kere daha dinleyin. En sonunda, istediğiniz ses bölümünü kesin olarak elde edin. Şimdi elde ettiğiniz sayıları pratikte nasıl kullanacaksınız? Bunları bir kenara mı not etmek lazım? Hayır! Böyle bir kullanım için, ses bloklarını bilgisayardan aklında tutmasını isteyebiliriz. Speech-Basic'te bu tip blokların başlangıcını ve bitişini aklında tutabilmesi için aynı isimle anılan bir komutu mevcut. Bunun kullanımı ise:

BLOCKx (FROM başl.adr.
(TO BİT.adr.)) (string)

Burada, başlangıç ve bitiş adreslerini demin belirlemiştik. String ise, 8 karakterden oluşan ve bu bloğu tanımlayan kısa bir açıklama. Blok numarasını belirleyen x'ise 1 ile 31 arasında bir sayı olabilir. 30 numaralı blok için daha evvelden, fonksiyon tuşlarının tanımlandığı bölgenin adresi ile belirlenmiş durumda. 31 numaralı blok için ise blok tanımlama tablosunun başlangıç ve sonuç adresi tanımlanmış durumda. Sıfır numaralı blok, gerçekte varolmasına rağmen, kullanıcı tarafından değiştirilemiyor. Bunun görevi, en son BLOAD işleminin başlangıç ve bitiş adresini akılda tutmak.

FROM başl.adr. TO bit.adr. parametrelerini gerektiren bir yerde, bunun yerine belirlenmiş bir blok bölgesini kullanabiliriz. Bunun için sözkonusu parametreleri BLOCKx gibi bir kullanımla özetleyebiliriz. Speech-Basic başlangıç ve bitiş adreslerini otomatik olarak blok tanımlamasından alır ve kullanır. Bu kullanım, PLAY, RECORD, BLOAD, BSAVE, MON ve BLOCK komutunun kendisinde geçerlidir. Mesela, buna göre şöyle bir satır yanlış olmaz:

BLOCK 1 BLOCK 0 "Merhaba"

Böylece, yeni yüklenmiş olan, "Merhaba" adlı ses örneğini BLOCK 1'e saklar ve BLOCK 0'ın içeriğini başka bir yerde saklayarak daha sonraki işlemler için kullanılmak üzere serbest hale getiririz. Gene benzer şekilde:

BSAVE"f-tuşları",8 BLOCK 30

kullanım olarak hatalı değildir ve

fonksiyon tuşlarının tanımlanmasını, daha sonra kullanılmak üzere disket-te saklamaya yarar. Blok tablosunu tümüyle silmek için şu döngüyü kur-mamız gereklidir.

```
FORx=1 T529:BLOCKx
"":NEXTx
```

Burada, başlangıç ve bitiş adresleri verilmediği için, bunların yerine otomatikman kullanılabilecek en düşük adres (HIMEM)'ile, kullanılabilecek en yüksek adres (SFFF8) oturtulur.

Blok tablosunu görmek istediğinizde, MAPx-y komutunu kullanmanız gerekir. Buradaki x ve y parametrelerinin kullanımı, LIST komutundaki kullanımın olduğu gibi aynısıdır. Örnek olarak, MAP 10, MAP 10-20, MAP-10 verebiliriz. Altı numaralı fonksiyon tuşu kendinden BLOCK komutu ile tanımlanmıştır. (Tabiî, bu tanımlamayı siz de değiştirebilirsiniz.) Şayet F6 tuşundan sonra (RETURN) tuşuna basarsanız, tüm blok tablosunu görürsünüz. Çünkü, BLOCK komutu, parametresiz olarak kullanıldığında MAP 0-31 komutu ile özdeştir.

BAZI ÖZEL EFEKTLER

Örneklediğimiz tüm sesleri izole edebildik mi, ses efektleri elde etmek artık çocuk oyuncağı gibi bir şey. Zaten, sözkonusu efektler program modunda bir şeyler yapmak gereksinimini doğuran tek neden. Diğer tüm işlemleri, direkt modda yapmak daha akıl kârı. En basit olarak elde edilen efekt RAP veya BREAK efekti, yani bir kelimenin kesik kesik tekrarlanması. Bunun için önce kelimenin kendisini (2), bunun içinde yeraldığı cümleyi (1) ve o kelimeden geri kalan kısmını (3) izole etmek ve belirlemek lazımdır. Daha sonra en basit şekliyle 1,2,2,3 sırasıyla ses geri verildiğinde sözkonusu efekt elde edilir. Geri verme sırasında, sesin hızıyla da oynanırsa SCRATCH efektleri elde etmek mümkün.

Bir cümlenin son kelimesini belirleyip, cümlenin arkasından bu kelimeyi tekrarlı ve azalan bir ses şiddetiyle geri verirsek ECHO veya HALL efektleri elde edilebilir. Burada, ufak bir aksaklık keşfediyoruz. Ses geri verilirken her bir örnekten bir diğerine geçilirken ekran bir kere göz kırpıyor ve geçiş bölgesi yarattığı hafif parazitle kulak tırmalıyor. Bu durumda yardımımıza EXEC komutu yetişiyor. EXEC sadece parametre kullanan bir komut. Parametreler bir karakter string'inin içinde yeralıyor ve çeşitli harf ve rakamlardan oluşuyor. Her bir harf bir komutu simgeliyor. Örnek verecek olursak, aşağıdaki program;

```
10 PLAY BLOCK 1: SPEED 1
20 PLAY BLOCK 2
30 PLAY BLOCK 3
40 PLAY BLOCK 4

EXEC"s1p1p2p2p3@"
```

olarak toparlanabiliyor. Tabiî ki burada string'in işlenmesi programın yürümesinden daha hızlı olarak gerçekleşiyor ve sesin geri verilmesi sırasında ekran daima kapalı kalıyor.

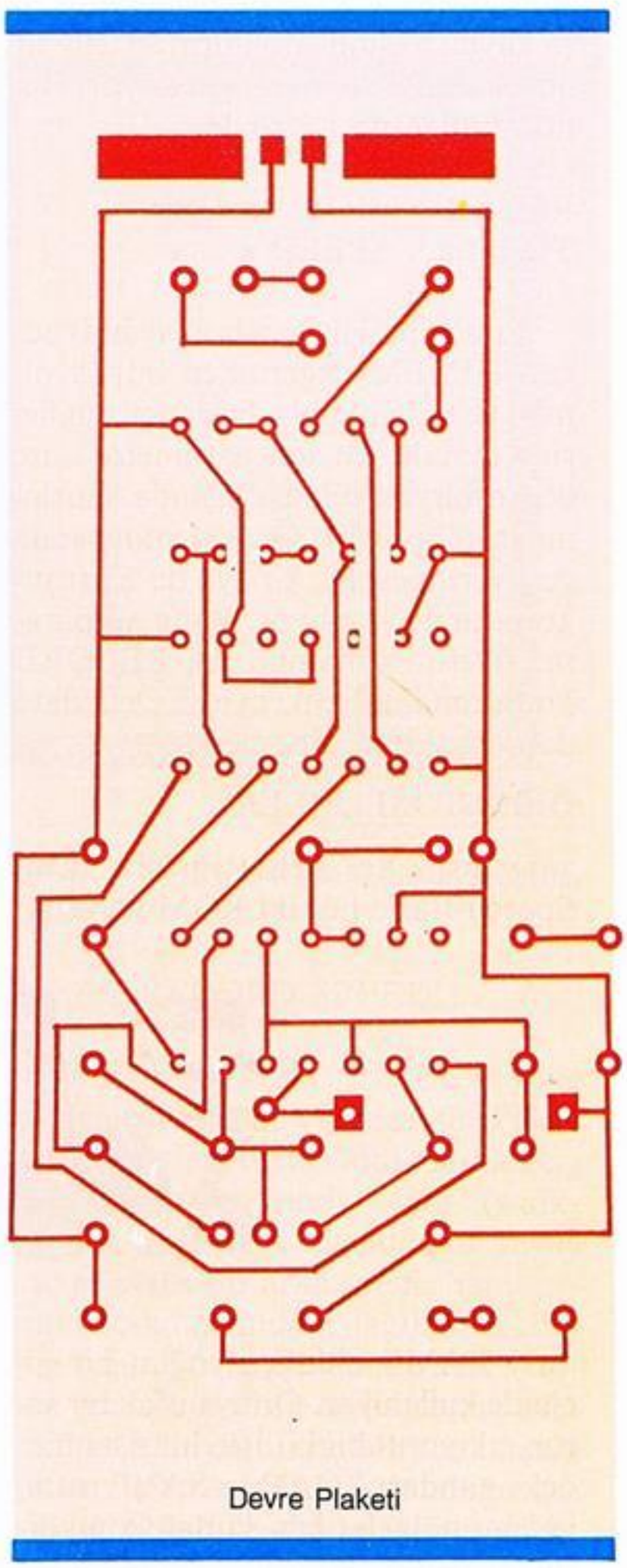
Şimdiye kadar anlattıklarımızı tam anlamıyla takip ettiyseniz, Speech-Basic'in nasıl çalıştığını ve neler yapılabileceğini anlamış olmanız gerekir. Fakat gene de, aşağıda yeralan (oldukça sıkıcı bir işlem ama ne yapalım), komutların kullanımını anlatan bölümü okumanız gerekiyor. Ayrıca, şimdiye kadar bahsedilmemiş birkaç komut ile ilgili açıklamaları da bulacaksınız aşağıda.

KOMUT LİSTESİ

Aşağıda, tüm komutlar yaptıkları işlerin benzerliği altında gruplanmıştır. Anlatım sırasında kullanılan büyük harfler komutların kendisi, küçük harfler ise bunlara ait parametrelerdir. Parantezler, sadece parametrelerin kullanılmama sırasını göstermek için konmuştur. Kullanılmayan parametrelerin yerine bilinen bir değer kullanılır. "Str" diye bahsedilen şey karakterlerden oluşmuş bir string'i temsil eder. Blok adı ile anılan şey belli bir hafıza bölgesidir ve üç türlü belirtilebilir:

1- FROM başl.adr. TO bit.adr.
2- başl.adr. , bit.adr.
3- BLOCK x

Büyük 'a' harfi (A), iki byte'lık bir tamsayıyı (adres) simgeler. Büyük 'n' harfi ise (N), bir byte'lık bir tamsayıyı temsil eder. Dv kısaltması, alet



Devre Plaketi

numarasıdır (device number).

Disketle yapılan tüm bilgi taşıma işlemleri, 127 no.lu mantıksal dosya kanalını kullanırlar. Bu kanalın bir özelliği, RUN/STOP + RESTORE işlemi sırasında otomatik olarak kapanmasıdır. Bu dünya kanalını kendi programlarınızda kullanamazsınız.

LISTING sırasında, (SHIFT) tuşu ile bekletme yapmak mümkündür.

Her yerde, onaltılık ve ikili sistemde sayılar kullanmak mümkündür. (PRINT $A000, x = $FCE2*%011011). Onaltılık sayılar, önlerinde yeralan '$' dolar işareti ile ayırdedilirler. İkili sayıların önünde ise '%' yüzde işareti bulunmak zorundadır. Onaltılık sayıların kullanımında karşınıza çıkabilecek bir sorundan bahsedelim. İçinde 'DEF' dizilimi bulunan onaltılık sayıyı 'D EF' gibi ayrı olarak yazın. Sayı olarak bu son kullanım gene geçerli olacaktır. Fakat,

Basic yorumlayıcısının, bunu 'DEF' komutu olarak yorumlayıp, ona ait token'ı (Basic yorumlayıcısının kendi komutlarını ayırdetmek için kullandığı kodlama) yerleştirmesi önlenecektir. Bu kurala uymadığınız takdirde, hata mesajı ile karşılaşırsınız.

IF...THEN komutları ile kurulmuş şartlı programlama satırlarında, THEN komutundan sonra yeni komutlardan biri gelecekse iki nokta üstüste ile ':' ayrılmış olması gereklidir. Yoksa, yazılımızın doğru olsa bile, hata mesajı ile karşılaşırsınız.

A) BAZI TEMEL KOMUTLAR

RESET-SID çipini, VIC çipini ve CIA çiplerini siler. Kernal'ın (işletim sistemi) sistem vektörlerini ilk değerlerle yükler. RESET komutu, her BASIC programın ilk komutlarından biri olmalıdır.

BASIC-Speech-Basic yardımcı programını terketme ve normal BASIC'e dönme. Bu komutu verdiğinizde "are you sure?" (emin misiniz?) sorusu ile karşılaşırsınız. Cevap olarak 'Y' (=yes) verirseniz, komut yürütülür. Başka herhangi bir cevapta, komut yürütülmez ve normal işlemler devam eder. Şayet, bu komut bir program satırının içinde kullanılmışsa bu soru sorulmaz ve komut doğrudan yürütülür. Bu arada, BASIC'in LOMEM (BASIC işlemleri için kullanabileceği en düşük adresli hafıza) değişkeni tekrar $0800 adresine oturtulmaz ve programın yürümesi kesilmez, sadece yeni eklenmiş komutların yürütülmesi kesilir.

HELP-Speech-Basic'e özel tüm komutları listeler. HELP* şeklinde bir kullanım, normal BASIC V2 komutlarını listeler.

B) DİSKET KOMUTLARI VE ÖZEL KOMUTLAR

KEY (n,str)- Fonksiyon tuşlarına bazı işlevler yüklemek. Bu komutu parametresiz kullandığımızda, tüm tuşlara atanmış işlevleri, ekranda edit edilebilecek bir şekilde görürüz. Eksi işareti "-" hiçbir tuşun tanımı sırasında kullanılamaz. Çünkü, bu işaret RETURN tuşunun kodunu ($0D) temsil etmektedir. Maksimum 15 karakterden oluşmuş tanımlar kullanılabilir. Speech-Basic'i ilk kullanmaya başladığınızda, programın kendisinden bazı tanımlar bu tuşlara yüklenmiştir. Bunlar:

F1 - RUN
F2 - LIST
F3 - PLAY
F4 - HEAR:RECORD
F5 - HELP/DISPLAY KEYS
F6 - BLOCK
F7 - DIRECTORY
F8 - DISK-STATUS

Aslında, normal çalışma sırasında, F5 tuşunu 'SPEED' komutu ile yüklemek oldukça uygun olur. Diyelim ki sözkonusu değişikliği yaptınız ve Speech-Basic'inizin artık böyle çalışmasını istiyorsunuz. Yapacağınız tek işlem Speech-Basic'in yeni versiyonunu diskete saklamak. Bunun için:

BSAVE"SPEECH-BASIC",8
FROM $0801 TO $1800

komutunu kullanın.

MEM- O sıradaki hafıza bölüşümünü rapor eder. Rapor aşağıdaki örnekteki gibi görünür:

BASIC : $1801 to $1803
SOUND : $a000 to $fff8
KEYS : $0c6f to $0cef
BLOCKS: $165b to $17db

DISK(str(,dv))- Disk kanalı ile ilgili işlemleri yürütmek. Bu komut parametresiz kullanıldığında, disket sürücüsünün durumunu ekranda gösterir. Eğer komut stringini kullanırsanız, bu disket sürücüsüne gönderilir. Fakat disket sürücüsünün mesajı alınmaz. Böylece istenirse, program içinde INPUT komutu ile hatanın veya durumun daha ayrıntılı işlenmesine olanak sağlanır.

DIR(str(,dv))- Disketin içeriğini görme. Parametresiz kullanıldığında disketin tüm içeriği gösterilir. Gönderilen parametre, dosya adı ve tipine göre seçim yapılmasını sağlayan bir maskedir. Bu maskenin kullanımının detayları için '1541 Disket Sürücüsü Kullanım Kılavuzu'na bakın.

BLOAD"prog.ismi"(,dv)(block)- İsmi verilmiş dosyayı 'dv' numaralı aletten hafızaya yüklemek. Blok kullanılırsa, yükleme işlemi burada belirtilen adresten başlar. Şayet kullanılmazsa, dosya kendi belirlediği adrese yüklenir. Yükleme işlemi dosya bittiğinde veya (belirtilmişse) bitiş adresine varıldığında biter. BLOAD komutunun LOAD komutundan bir başka farkı ise bunun normal bir BASIC komutu gibi yürütülmesi. Yani LOAD'daki gibi yükleme işlemi bittikten sonra program baştan başlamıyor. Yüklenen dosyanın başlangıç ve bitiş adresini belirlemek istiyorsanız BLOCK 0'ı kullanabilirsiniz. BLOAD komutu daimi RAM'e yükler (zaten istese de ROM'a yükleyemez, burada belirtilmek istenen bütün yükleme işlemi sırasında RAM konfigürasyonunun kullanıldığı, yani $D000-$DFFF adresleri arasını yüklerken bile bilgisayar 'crash' etmiyor, çünkü dosya buranın üç (!) kat altındaki RAM bölgesine yükleniyor). BLOAD komutunun kullanımına bazı örnekler vermek istersek:

BLOAD"s.test"
BLOAD"s.test",8 FROM $A000
BLOAD"s.test",8 FROM $A000,
$8000
BLOAD"b.test",8 BLOCK 31:REM
blok talosunu yüklemek.

BSAVE"prog.ismi"(,dv)(block)- Başlangıç ve bitişi belirtilen bölge verilmiş isim altında dv numaralı alete saklanır. Burada dosyaları ayırdedebilmeniz için bir önerimiz var: Diyelim ki aynı örnekler için ses bilgilerini ve bunların içindeki başlangıç ve bitişlerini belirtilen blok tablosunu saklamak istiyorsanız. Her iki dosyayı aynı isimle saklayın; tek farkla: Ses bilgilerinin başına 's.' (ses), blok tablolarının başına 'b.' (blok) ekleyin. Böylece, ileride oluşabilecek karşılıkların önüne geçersiniz. BSAVE komutuna ait birkaç örnek:

BSAVE"s.test"
BSAVE"s.test",8 FROM $A000
BSAVE"s.test",8 FROM $A000,
$8000
BSAVE"b,test",8 BLOCK 31:REM
blok tablosunu saklamak

DIGITIZER

C) SES KONTROL KOMUTLARI

HEAR(SPEED n)- Örneklenen sesin aynı anda duyulmasını sağlar. Burada dört değişik örnekleme seviyesine karşılık olarak dört değişik ses şiddeti seviyesi ve dört değişik rengin belirlenmesi gerekmektedir. Renk değerleri $D020, ses şiddeti değerleri $D418 adresli registere yazılmaktadır. Dinleme işlemini bitirmek için boşluk tuşuna basın.

RECORD (block) (SPEED n)- HEAR komutu ile aynı biçimde çalışır. Tek fark, dinlenen sesin aynı zamanda hafızaya da yazılmasıdır. Blok dolmadan, ses kayıt işlemi kesilmek istenirse RESTORE tuşuna basmak yeterlidir. Direkt moddayken bu işlem yapılırsa, kesilme adresi 'stopped at xxxxx.' mesajı ile belirtilir. Örnekleme hızı 1 ile 40 arasında değişen bir tamsayıdır. Burada 1, en hızlı kaydı, 40 ise en yavaş kaydı sağlar. Örnek olarak:

```
RECORD
RECORD FROM $A000 TO $D000
RECORD SPEED 1
RECORD BLOCK 1 SPEED 10
```

PLAY(block)(SPEED n)- RECORD işleminin tam tersidir. Blok içindeki bilgiler, ses ilgisi olarak yorumlanıp, karşılık gelen ses şiddet değerleri ve renk değerleri kullanılarak geri verilir. En yüksek geri verme hızı olarak '0' kullanabiliriz. Yani sesi kayıttan daha hızlı geri verebiliriz.

VOLDEF (n,(n,(n,(n))))

COLDEF (n,(n,(n,(n)))) Dört değişik örnekleme seviyesine karşılık gelen ses şiddet ve renk değerlerini belirlemek. Digitizer devresinden elde edilen değerler doğrudan ses ve renk kanallarına yazılmaz. Bu işlemden önce gelen bilgiler, sizin bu komutlarla belirlediğiniz tablolar vasıtasıyla işlenir ve öyle kullanılır. İki değişik örnekle, VOLDEF ve COLDEF komutlarının parametrelerinin etkilerini inceleyelim:

```
VOLDEF0,1,2,3:COLDEF0,11,12,
15: REM alçak sesli ve gri tonları
VOLDEF0,5,10,15:COLDEF7,2,5,
6: REM yüksek sesli ve çok renkli
```

D) DAHA İLERİ SEVİYEDEKİ KOMUTLAR

BLOCK n (block)(str)- Bir blok, tanım olarak, başı ve sonu belirlenmiş ve bunlarla sınırlanmış hafıza bölgesidir. Birçok komutun kullanımında yeralan (block) parametresi ya 'FROM başl.adr. TO bit.adr.' biçiminde, ya da FROM ve TO komutları kullanılmayarak yeni 'başl.adr., bit.adr.' biçiminde kullanılabilir. Başlangıç ve bitiş adresleri genelde ayrı ayrı kullanılmadığı için BLOCK komutu geliştirilmiş ve her iki adresi tek komutta işlemeye olanak vermiştir. Bitiş adresi olarak tanımlanan sayının, başlangıç adresinden büyük ve $FFF9 adresinden küçük olması gerekmektedir.

BLOCK komutu parametresiz kullanıldığı zaman KEY komutu gibi tüm blok tanımlamalarını ekranda listeler.

Blokların sayısı 0'dan 31'e kadar uzanabilir. Fakat 0'ıncı blok özel bir işlem için ayrılmış olduğundan bu blok'un kullanıcı tarafından tanımlanması mümkün değildir. Bu blok, en son yapılan BLOAD işleminin başlangıç ve bitiş adresini aklında tutar. 30 numaralı blok, fonksiyon tuşlarının başlangıç ve bitiş adresini, 31 numaralı blok, blok tablosunun bulunduğu bölgenin başlangıç ve bitiş adreslerini aklında tutar. Gerektiğinde bu bölgeler diskette saklanmak istendiğinde (gerçekten de saklamak gerekiyor bu bölgeleri) sözkonusu adresleri aklınızda tutmanız gerekmiyor, 'BLOCK' komutu bu işi sizin için hallediyor. Fakat isterseniz 30 ve 31 numaralı blokları kendi istediğiniz biçimde yeniden tanımlayabiliyorsunuz. Şayet böyle bir işlem sonucu, sözkonusu değerleri kaybettiyseniz bunları MEM komutundan elde edebilirsiniz.

Burada parametre olarak verilen str adlı string ise, sekiz karakteri aşmayan ve yalnızca blok tablosu istendiğinde ortaya çıkan, blokların içeriği ile ilgili kısa bilgileri saklasın diye bırakılmış bir açıklama bölgesidir. Blok tablosunu RESET'lemek için (tümüyle silmek, ilk ve temiz haline getirmek için):

```
FOR n=1 TO 29:BLOCK n"":
                    NEXT n
```

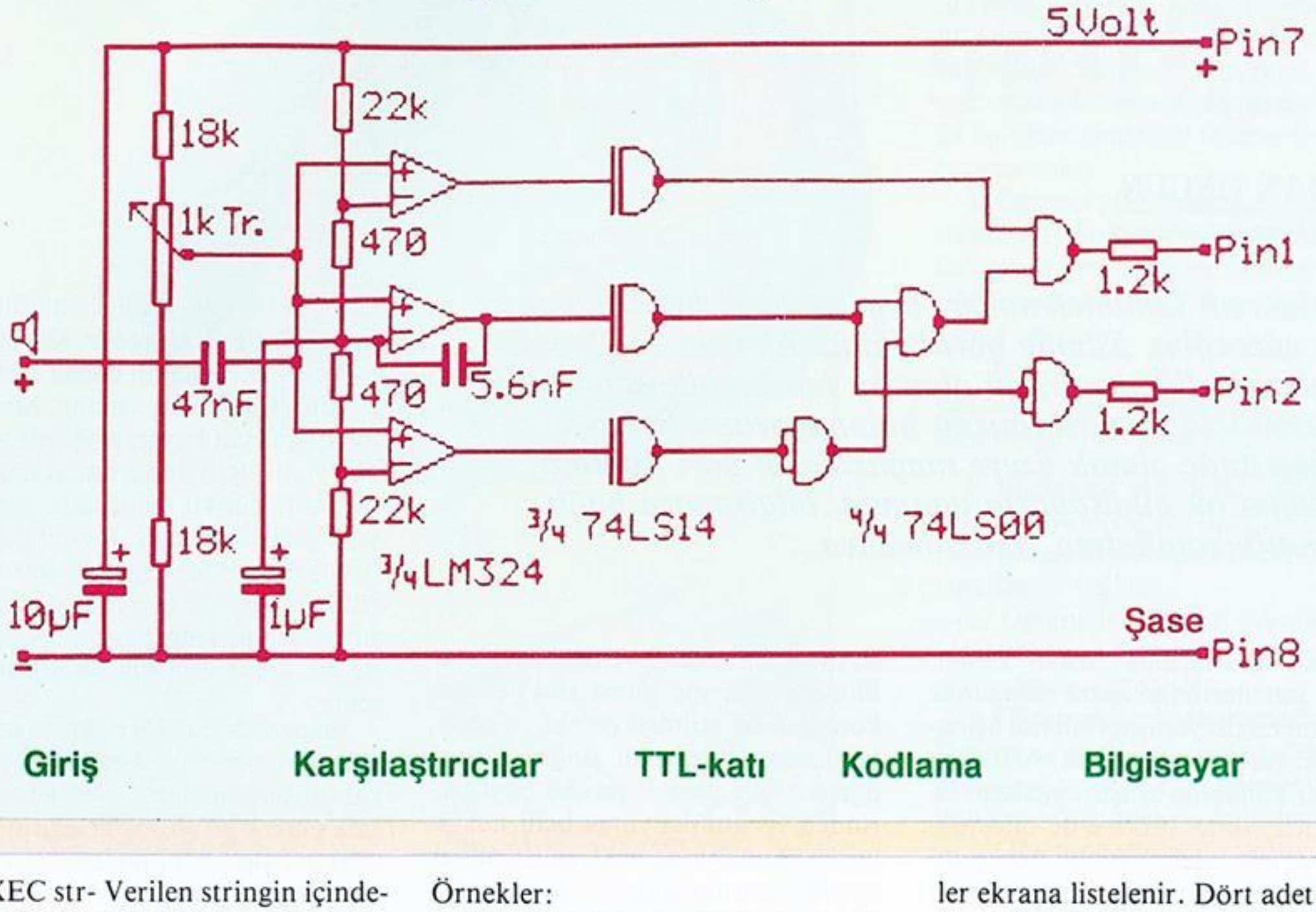
satırını vermeniz yeterlidir. Bu komuta ait birkaç örnek isterseniz:

```
BLOCK 1 "":REM 1 numaralı
blok'u silme
BLOCK 1,$A000:REM bitiş adresi
olarak $FFF8 kullanılır
BLOCK 1,$A000,$8000:REM aşağı-
dakinin kısaltılmışıdır.
BLOCK 1 FROM $A000 TO $8000
BLOCK 1 BLOCK 0:REM 0 numa-
ralı blok'un içeriğini saklamak
BLOCK 1 BLOCK 0:REM 0 numa-
ralı blok'un içeriğini saklamak ve
isimlendirmek.
BLOCK 1 TO $8000:REM
YANLIŞ!!!
```

map(x(-(y)))/(-y)- Blok tablosunu edit edilebilir şekilde listeler. MAP komutunun arkasında yeralan parametre listesi sizde korku yaratmasın. Aslında parametreler göründüğü kadar korkunç değil, tam aksine LIST komutunun kullanımı ile olduğu gibi aynı bu parametrelerin kullanımı.

HIMEM A- Basic için kullanılabilecek hafızanın sonunu (üst sınırını) belirliyor. Böylece, aynı zamanda ses bilgileri için kullanılabilecek alt sınırı da belirlemiş oluyoruz. 'A' değeri, o sırada hafızada bulunan programın son adresinden büyük ve $A000'dan küçük olmak zorundadır. HIMEM komutu yürürken CLR işlemini de yürütür. Örnekler:

Digitizer devre şeması



EXEC str- Verilen stringin içindeki karakterleri birçok ses komutunun kısaltılmışı olarak değerlendirir. Bunlardan başka, ses şiddeti ve renk değerleri de bu string'in içinde belirlenebilir. Kısaltılmış olarak yeralan komutlar şunlardır:

p w ; w numaralı blok'u çalma
s w ; hızı belirleme
w w ; w/60 saniye beklemek
v w,w,w,w ; ses şiddet değerlerini belirlemek
c w,w,w,w ; renk değerlerinin belirlenmesi
str ; str diye belirtilen değişkenin de yürütülmesi
@ ; EXEC işleminin bitirilmesi

Diğer karakterlerden herhangi biri EXEC komutunun yürümesinin kesilmesine neden olur. Her bir komut birbirinden, iki nokta üstüste ':' işareti ile ayrılabilir, fakat şart değildir, istenirse kullanılmayabilir. Parametreleri simgeleyen 'w' bir sabit veya bir değişken olabilir.

DİKKAT: EXEC işlemini bitiren '@' işaretini mutlaka kullanın. Yoksa, programınız istenmeyen komutları (!) yürütebilir.

Örnekler:
EXEC "s1:p1:p1:p2:p3:p2:p3: "
EXEC 's1v0,5,10,15p1 c$"
EXEC a$+"@":REM işleminin sıhhatli olarak bitirilmesini kesinleştirme
EXEC A$+b$"@"

E) DİĞER KOMUTLAR

HEX ve DEZ- RESTORE tuşu ile yapılan durdurma işlemleri sırasında verilen mesajın içindeki sayının hangi sistemde verileceğini belirler. ('stopped at xxxxx') HEX komutu, hekzadesimal yani onaltılık sistemi, DEZ komutu ise desimal yani onluk sistemi seçer.

SCREEN flag- Ekranın, ses ile ilgili işlemler sırasında (HEAR, RECORD, PLAY ve EXEC) kapanıp kapanmaması gerektiğini belirler. Normalde bütün bu işlemler sırasında ekran kapalıdır ve bu durum ses kalitesini kısmen yükselmektedir. Parametre olarak belirtilen 'flag' ON ve OFF olabilir. Bundan başka, bu komutun hemen arkasından gelecek bir matematiksel önerme de aynı işi görür. (Önermenin değerinin sıfır veya sıfırdan farklı olması.)

MON (block)- Ses monitoru. MON komutuyla, örneklenen bilgiler ekrana listelenir. Dört adet örnek bir byte'ı oluşturur. Her bir değer, VOLDEF'te verilen ses şiddeti değerleri ile duyulabilir sese, COLDEF'te verilen renk değerleri ile gözün algılayabileceği renk bantlarına dönüştürülür.

Değerler, üzerine gidip yazmak suretiyle değiştirebilir. Değişiklikler yapıldıktan sonra RETURN tuşuna basmak gerekmektedir. MON komutuyla, kısa süreli gürültülü aralıkları düzeltmek mümkündür. MON komutun en çok, Digitizer devremizin ayarını, kontrolunu yapmamıza yarar. Örnek verecek olursak:

MON FROM $0400 TO $0401 komutunun sonucu; $0400:42444244
İyi ayarlanmış: $B000:
1111223444332211 1
Yüksek ayarlanmış: $B000:
1111144444441111
Alçak ayarlanmış: $B000:
2223233433222212

Şekil-4'te, digitizer devresinin komple devre şemasını bulacaksınız. Yeni digitizer'ınızla yapacağınız çalışmalar sırasında bol bol eğlenceler diliyoruz. □

Bilgisayar destekli eğitime geçerken

OSMAN ONGUN
Eğitim Programları Yazarı

"Muhterem hanımefendiler, beyefendiler, bir dakikanızı rica edeceğim. Elimde gördüğünüz Alaman ve Capon teknik işbirliğinin ürünü olan bu harika bilgisayarın tanıtımı için huzurlarınızda bulunuyorum. Reklâm mahiyetinde olmak üzere mağazaların yarı fiyatına verdiğim bu bilgisayarın yanında, bilgisayara bağlı otomatik zayıflatma cihazı bedava..."

Boğaz vapurlarında cüzdan, kalem, tarak satıcılarından âşina olduğunuz bu sesin bilgisayar işportalarını açması belki yarından da yakın. Artık bilgisayar kullanımı araştırmacıların tekelinden çıktı. Türkiye'de bile yüzbinlerce aile bilgisayarla tanışmış durumda. Belki şimdilik daha çok oyun aracı olarak kullanılıyor; ama okullar bilgisayar donanımına kavuşturuldukça, müfredat programlarındaki iyileştirmelerle birlikte eğitim yazılımları arttıkça bilgisayarın eğitim dünyamıza getireceği atılımın boyutları daha iyi anlaşılacaktır.

Gelişmiş ülkelerle aramızdaki teknolojik açığın kapatılması açısından büyük önem taşıyan "Eğitimde Bilgisayar" konusuna iki açıdan yaklaşabiliriz:

1. Ortaokuldan başlayarak gençlerimize bilgisayar programlama dillerinin öğretimi.

2. Anadili, yabancı diller, matematik, fen ve doğa bilimlerinin öğretiminde bilgisayar desteğinden yararlanma.

Bu söyleşimiz, konunun ikinci yönüne ağırlık verecektir.

EĞİTİM VE ÖĞRENME

Eğitim, insan davranışlarında olumlu değişiklikler yapmayı amaçlar. Öğretim, eğitimin gerçekleşmesi için önemli bir araçtır. İki yaşına gelmiş bir çocuğun kendi kendine yemek yemesinin sağlanması bir eğitim sürecidir. Bu eğitimin gerçekleşmesi için çocuğa, kaşık ve çatalın kullanılması, tabağına alacağı yemeği ayarlayabilmesi, çiğneme süresi gibi pek çok konunun öğretilmesi gerekir. Çocuk, bunlardan bazılarını doğru olarak öğrense bile yemek yerken büyüklerinden yardım istiyorsa belli noktalarda öğrenme gerçekleşmiş; ancak eğitim başarılamamıştır. Çocuk, görgü ve sağlık kurallarına uygun olarak kendi kendine yemek yemeye başladığında bu konudaki eğitimini tamamlamış, davranışlarında istenilen değişiklik sağlanmış olur. Bu kolay götüren eğitimin bile yıllar aldığını ana-babalar çok iyi bilirler. Örnek üzerinde biraz düşünmek, eğitimin ne denli güç ve karmaşık bir süreç olduğunu görmemizi sağlayacaktır.

Okullar, kitlelerin eğitimi için açılır. Ne var ki elli kişilik bir sınıfta elli ayrı dünya vardır. Bu elli çocuğun yetenekleri, öğrenme hızları ve istekleri farklıdır. Öğretmenin ortalama bir düzeyde hazırlayıp kürsüden anlattığı dersi bazı öğrenciler zevkle dinlerken, bazıları sıkılmakta, bir bölümü ise konuyu kavramakta bile güçlük çekmektedirler. Öğrenmenin bireyselliğini kanıtlayan bu durumu hepimiz biliriz. İlkokuldan başlayarak eğitim süresinin sonuna kadar öğrenciler arasında büyük farklılıkların ortaya çıkması başarısızlık yüzünden yaşanan intihar girişimleri, seçkin okullarda okuma özlemi, klasik eğitim yöntemlerinin günümüzün ihtiyaçlarına cevap verememesinden kaynaklanmaktadır.

Bir marangoz yaptığı yanlışlığı yeni malzemeyle düzeltebilir. Kaybı, yanlış kestiği kerestenin bedeli kadardır. Eğitimde harcanan zaman, bir başka deyişle, insan hayatı üzerinde yapılacak yanlışlığın düzeltilmesi mümkün değildir; çünkü yanlışlığın yapıldığı yaşa geri dönülemez. Bu noktada, gelişmiş ülkelerde yirmi üç yıldır sürdürülen bilgisayar destekli eğitim araştırmalarının sonuçları, ülkemiz için de bir umut ışığı olarak görünmektedir.

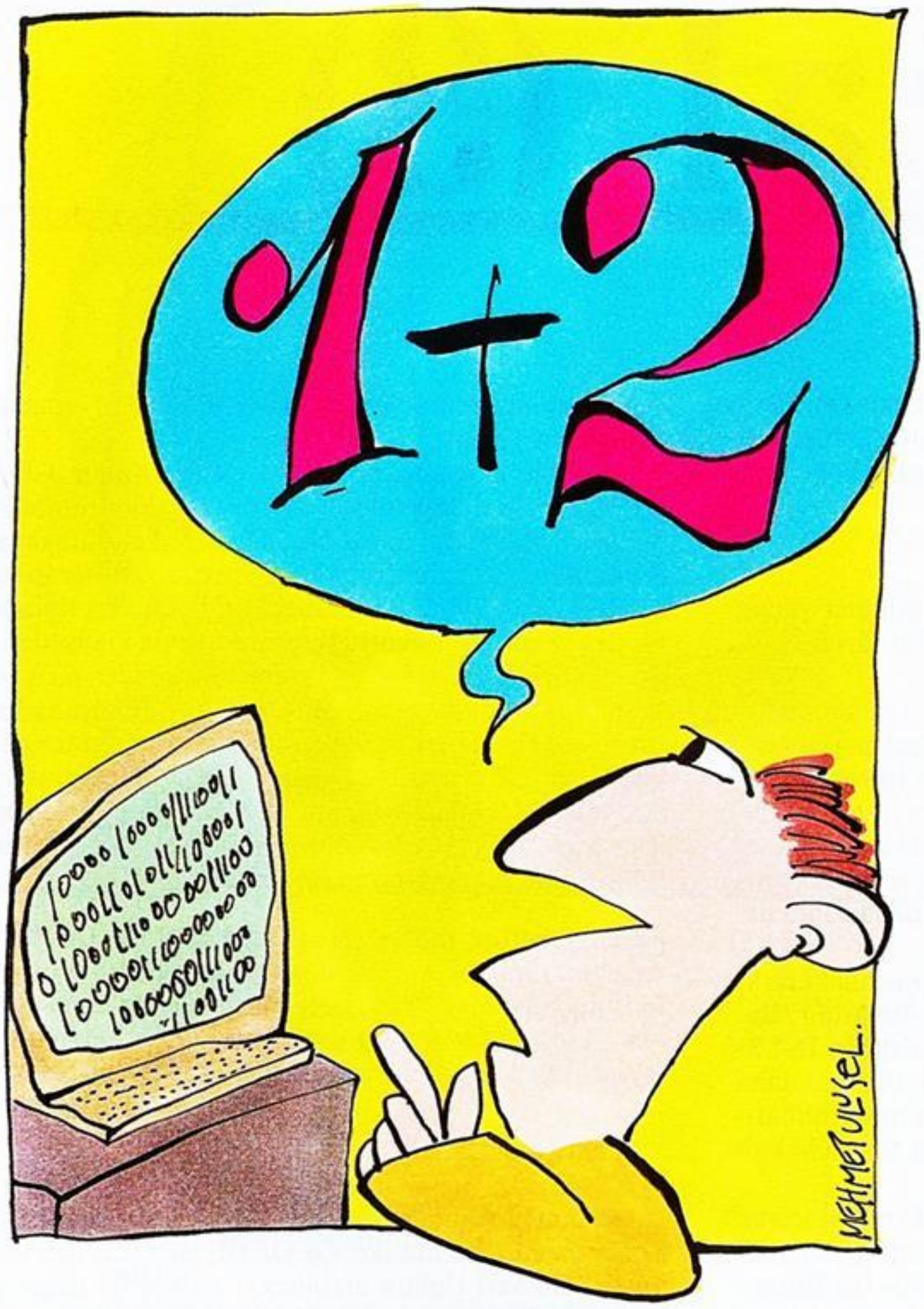
Bilgisayar da diğer eğitim araçları gibi öğretmene yardımcı bir araç olarak düşünülmelidir. Ne kadar gelişmiş olursa olsun hiçbir eğitim aracı öğretmenin yerini tutamaz. İnsan sıcaklığını taşıyan, öğretmenin çocuğa vereceği sevgi ve şefkati çocuğa iletebilecek bir bilgisayar henüz yapılmamıştır.

Bilgisayar, öğretmenin yükünü azaltan, onu bir yığın mekanik işten kurtarıp yaratıcılığa yönelten bir araçtır. Bilgisayarla desteklenen öğretmen, öğrencilerle daha çok ilgilenebilecek, bireysel farklılıklara göre yeni programlar geliştirebilecek, kısacası klasik eğitimin çıkmazlarından kurtulma şansını elde edecektir.

Burada akla hemen şöyle bir soru takılabilir: "Acaba yüz binlerce öğretmene bir anda bilgisayar programcılığı öğretilebilir mi?" Şunu hemen belirtelim ki öğretmenin bilgisayardan yararlanması için bilgisayar dillerini ve işletme prensiplerini bilmesine gerek yoktur. İyi tasarlanmış akış diyagramlarına uygun program kanavaları, öğretmene hazır olarak sunulabilir. Öğretmenin bu kanavalara göre program hazırlaması için daktilo makinesi kullanmayı bilmesi yeterlidir.

Eğitim programlarının hazırlanmasındaki aşamaları, TELETEKNİK A.Ş.'nin siparişi üzerine hazırladığım

Commodore



programlarda izlediğim yolu anlatarak örnekleyeceğim:

a) Hazırlık çalışmaları:

1. Öğretmen, müfredat programında öğrencilere kazandırılması istenen bilgi ve becerilerin bir dökümünü yaparak hangi konularda program hazırlayacağını belirler.

2. Bu konuların ne kadar sürede işleneceğini planlayarak her birim-zaman için bir program hesabıyla hazırlanması gereken program sayısını tespit eder.

b) Derslerin verilmesi:

1. Öğretmen, seçtiği konuda öğrencilere neleri öğretmek istiyorsa bunları, planına uygun olarak bilgisayara girer.

2. Bazı gözlem ve deneyleri kolay ve ucuz olarak ekranda gerçekleştirmek mümkündür. Öğretmen dersin verilişi sırasında yapacağı deneyleri, çizeceği grafikleri hazırlar.

3. Dersleri yazarken seslendiği öğrenci kesiminin bilgi ve kavrama düzeyine uygun bir anlatım yolu seçerek kullanacağı yöntem ve teknikleri belirler.

c.) Öğretilen bilgilerin denetlenmesi:

1. Öğretmen, programda verilen bilgilerin öğrenilip öğrenilmediğini denetlemek amacıyla hazırladığı yeterli sayıdaki çoktan seçmeli soruyu bilgisayara girer.

2. Hazırladığı soruların bütün ayrıntıları kapsamasına, geçerlik, güvenirlik ve objektiflik ilkelerine uygun olmasına dikkat eder.

3. Gerektiğinde ekrana getirileceğini düşünerek her sorunun ayrıntılı cevabını ve çözüm yollarını bilgisayara girer.

d) Eksik öğrenilen bölümlerin belirlenmesi ve yeniden öğretilmesi:

1. Öğrenci soruları yanıtlarken bilgisayar, hangilerini doğru, hangilerini yanlış yaptığını, hangi sorulara yanıt vermekten kaçındığını belleğinde saklayacak biçimde programlanır.

2. Sınavın bitiminde yanlış yapılan ya da yanıtlanmayan sorular yeniden ekrana gelir.

3. Öğrenci yine başarısız olursa sorunun dayandığı bilgiler, açıklamalar, çizim ve çözümler ekrana gelir. Böylece konuyla ilgili boşluklar giderilmiş olur.

BİLGİSAYAR DESTEKLİ EĞİTİMİN YARARLARI

1. Bilgisayar sabırlı bir tekrarlayıcıdır. Bu yönüyle geç öğrenen çocukların konuyu diledikleri kadar tekrarlamalarını sağlar.

2. Öğrenciler, iyi programlanmış bir dersi oyun oynar gibi isteyerek ve zevk alarak öğrenirler.

3. Öğrencilerin kavrayamadığı bölümleri ve niçin kavrayamadıklarını anında görmek ve dikkatlerini bu bölümlere yöneltmek kolaylaşır.

4. Öğrenme süresini kısaltır, öğrenilen bilgilerin kalıcı olmasını sağlar.

5. Bilgileri depolama ve gruplandırmayı kolaylaştırır.

6. Öğrencinin kendine güvenini ve cesaretini artırır.

7. Öğrencinin dikkat, kavrama, ilişkilendirme, karşılaştırma, zihinde cezalandırma, akıl yürütme yeteneklerini geliştirir.

8. Öğrencinin grup içindeki yerini, bilgi ve yetenek düzeyini belirler.

9. Öğretmenin yükünü azaltarak onun daha yaratıcı çalışmalara yönelmesini sağlar.

BİLGİSAYARLA ÇALIŞMAK ÇOCUĞU TOPLUMDAN KOPARIR MI?

Bu soru, bilgisayar almak isteyen pek çok veli tarafından sorulmaktadır. Üç yıldır bilgisayar destekli eğitim çalışmalarını sürdürmekteyim. Korkuların aksine bilgisayarla çalışma, öğrencilerim arasında canlı ve içten bir etkileşim sağlamakta, bu etkileşim ve dayanışma okul dışında da sürmektedir.

Sözlerimi bilgisayar destekli eğitimin yaygınlaşması dileğiyle noktalarken, hepinize teşekkür eder, saygılar sunarım. □

Pascal (3)

BURAK KİPER

Yazımızın geçen bölümünde PASCAL'ın bazı temel komutları ile, çıktılarımızı nasıl düzenleyebileceğimizi görmüştük. Bu yazımıza veri tiplerinin tanıtılması ile başlayacağız.

VERİ TİPLERİNE GİRİŞ

PASCAL'ı bilgisayar dilleri arasında popüler yapan özelliklerden biri de birçok ayrı türde bilgi türünü kullanabilmesidir.

PASCAL'da iki tür numerik veri tipi kullanılabilir: REAL ve INTEGER. Bu veri tiplerinde çalışırken aritmetik işlemlerden +, —, *, / ve eşitleme işareti olarak da ':=' kullandık. Bu aritmetik işlemcilere ek olarak INTEGER veri tipinde kullanabileceğimiz iki aritmetik işlemci daha bulunmaktadır. Bunlar DIV ve MOD aritmetik işlemleridir. Bu aritmetik işlemleri daha sonra detaylı olarak anlatacağız.

INTEGER veri tipi ile REAL veri tipi arasındaki fark, INTEGER veri tipinde sayıların tam sayı olmasıdır. Buna karşın REAL veri tipindeki sayılar kesirlidir. INTEGER veri tipinin REAL veri tipine göre kullanımı daha kısıtlıdır. INTEGER veri tipini daha çok programlarımızda herhangi bir olayın tekrarlanmasını saymada kullanırız.

REAL veri tipindeki sayılar bir desimal nokta içeren rakam grubundan oluşur. REAL veri tipindeki sayıları bilimsel gösterimlerde yazmak mümkündür. Bu durumda sayı bir tam sayı veya kesirli sayı ile başlar, bunun hemen arkasından 'E' harfi gelir. Bu harfi '+' veya '—' işaret, en sonunda da sayının kuvvetini içeren iki rakam olacaktır. REAL veri tipindeki sayıların geçerli ve geçersiz olabilecek örnekler bulunmaktadır.

Geçerli REAL sayılar	Geçersiz REAL sayılar
3.14	260 (. dan sonra sayı yazılmamış)
-0.0007	-.12198 (. dan evvel sayı yazılmamış)
+54321.0	.14 (. dan evvel sayı yazılmamış)
-34E-04 (=-0.0034)	-18.E-07 (-18. sayının REAL kısım hatalı)
-4.345E2 (=-434.5)	21E5 (.5 sayının tam kısmı hatalı)
1.3E+6 (=1200000)	.1234E5 (1.1234 REAL kısmı hatalı)

INTEGER veri tipi ile ilgili olarak PASCAL'da MAXINT standart belirteci tanımlanmıştır. Bu standard değişkenin değeri her PASCAL sisteminde farklıdır, kullanmakta olduğunuz PASCAL sisteminde bu standart değişkenin değerini öğrenmek içir WRITELN (maxint) komutunu yazınız.

Veri tiplerinden üçüncüsü CHAR veri tipidir. CHAR veri tipindeki değişkenlerde tek bir karakter tutabilirsiniz. Bu tipte tanımlanmış olan bir değişkene atayacağınız karakteri (') işareti içinde bulundurun. Programınızda READLN komutu kullanarak CHAR veri tipindeki bir değişkene bilgi okuttuğunuzda o anda klavyeden girilen karakter bu değişkene atanır. Klavyeden giriş sırasında boş karakter göndermek için boşluk tuşuna basınız. Bir CHAR veri tipindeki değişkene programınızda boş karaktere atamak istiyorsanız AD: = ' ' şeklinde yazın, eğer '' şeklinde yazarsanız bir hata mesajı ile karşılaşırsınız. Aşağıda CHAR veri tipinde bir değişkenin ve sabitin tanımlanması yeralmaktadır.

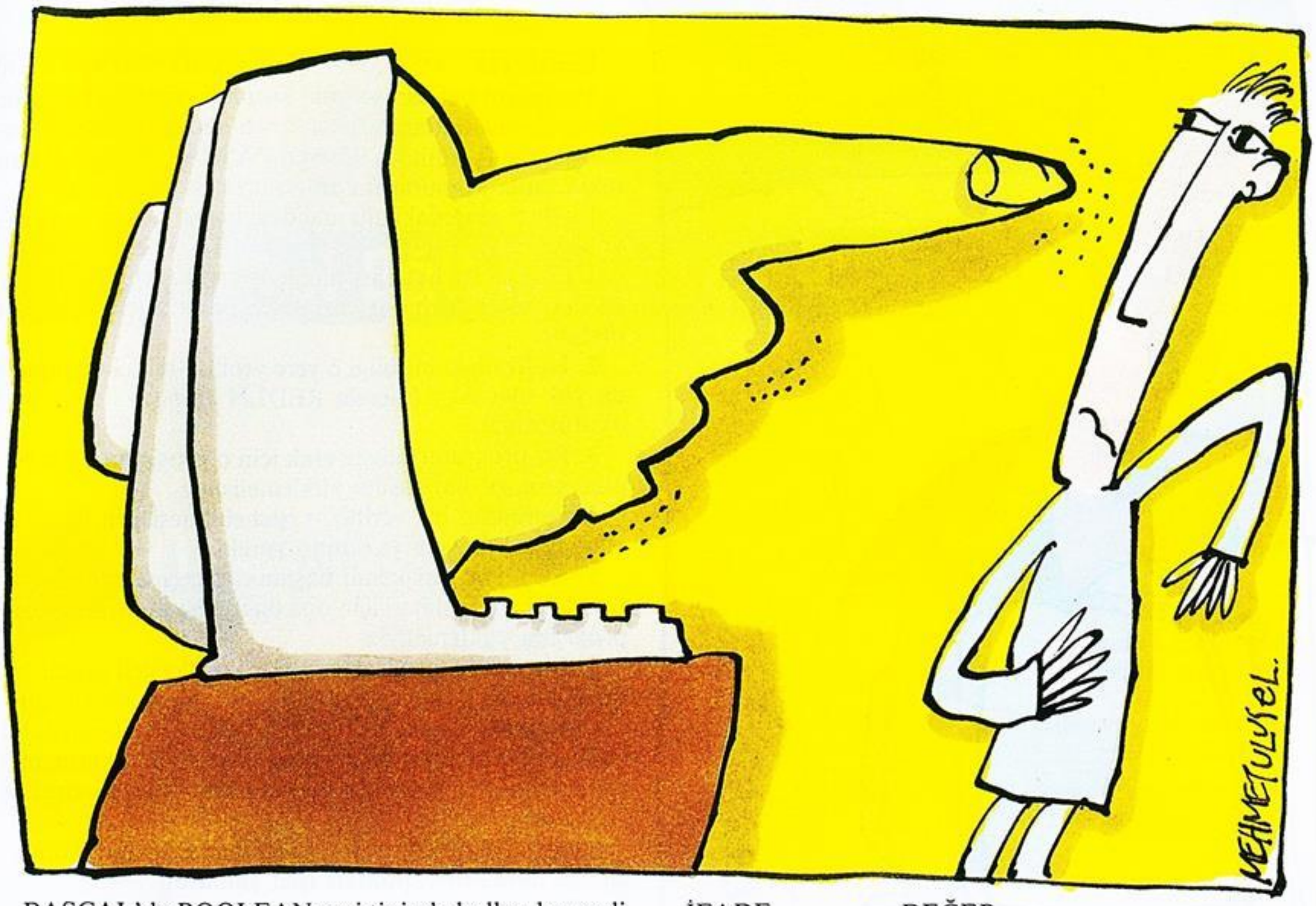
```
CONST

    soru = '?';

VAR

    har, ad: CHAR;
```

Dördüncü standart veri tipi BOOLEAN'dır. Bu veri tipine bu ad matematikçi GEORGE BOOLE'den verilmiştir. Bu veri tipinin alabileceği sadece iki değer bulunmaktadır. Bu değerler TRUE ve FALSE'dur.

Bu veri tipi, yazımızın ilerki bölümlerinde göreceğimiz kontrol komutlarında kullanılmasında bize lazım olacaktır. Mesela I değişkeninin değeri J değişkeninin değerine eşittir gibi komut eğer her iki değişkenin değeri de birbirine eşitse doğrudur, aksi takdirde yanlıştır. Bu komutun doğruluk veya yanlışlığına bilgisayarın karar vermesi için bu iki değişkenin değerlerinin hafızaya konması gerekir. Bu karşılaştırmada sonucu BOOLEAN veri tipinde bir değişkene konduğunu düşünürsek bu değişken TRUE veya FALSE değerlerinden birini alır.

BOOLEAN veri tipinde kullanacağımız işlem işareti "=" (eşit'tir). örnek olarak J=I diyebiliriz. Eğer J ve I değişkenlerinin değerleri birbirlerine eşitse o zaman bu komut doğrudur. Eğer değişkenlerin değerleri birbirlerinden farklı ise bu komut yanlıştır. = işlemcisinin kullanımına örnekler aşağıda yeralmaktadır.

İfade	Değer
4=8	FALSE
3=3	TRUE
0=1	FALSE
2=2	TRUE



PASCAL'da BOOLEAN veri tipinde kullanılan yedi işlemci vardır. Biz şu anda bunlardan altısını tanıyacağız. Aşağıda bu işlemleri kullanarak yapılmış komutlar bulunmaktadır. Bu komutlara göre BOOLEAN ifadesinin alacağı değerler açıklanmıştır. I ve J komutlarda kullanılan değişkenlerdir.

İFADE	DURUM
I = J	I değişkeni J değişkenine eşittir.
I < J	I değişkeninin değeri J değişkeninden küçüktür.
I > J	I değişkeninin değeri J değişkeninden büyüktür.
I < = J	I değişkeninin değeri J değişkeninden küçük veya eşittir.
I > = J	I değişkeninin değeri J değişkeninden büyük veya eşittir.
I < > J	I değişkeninin değeri J değişkenine eşit değildir.

Bu işlemcileri REAL, INTEGER, CHAR veri tipleri ile de kullanmak mümkündür.

Aşağıda üstte verilen işlemcilerin komutlarda kullanımına örnekler verilmiştir.

IFADE	DEGER
4 < 6	TRUE
3 < > 3	FALSE
5 > 4	TRUE
1.2 < 2.4	TRUE
2.5 >= 1.2	FALSE
4.4. < = 4.4	TRUE

CHAR veri tipinde bu işlemcilerin kullanılmasında şu örnekler verebiliriz.

İFADE	DEĞER
' ' < 'A'	TRUE
'0' > 'V'	FALSE
'A' < 'Z'	TRUE

Şimdiye kadar gördüğümüz işlemciler'in birbirlerine göre üstünlük sıraları vardır:

* , / , DIV, MOD	İlk yapılacak
+ , —	
= , < , > , < = , > = , < >	Son yapılacak

İşlemcilerin sıralarına örnek:

5 + 2 < (1*3—2	İlk önce çarpma işlemi
3 + 2 > 7—5	İlk önce toplama ve çıkarma
12 DIV 4 = 2 + 2	İlk önce DIV
5 = 3 + 2	İlk önce toplama
3 < > 5*2	İlk önce çarpma

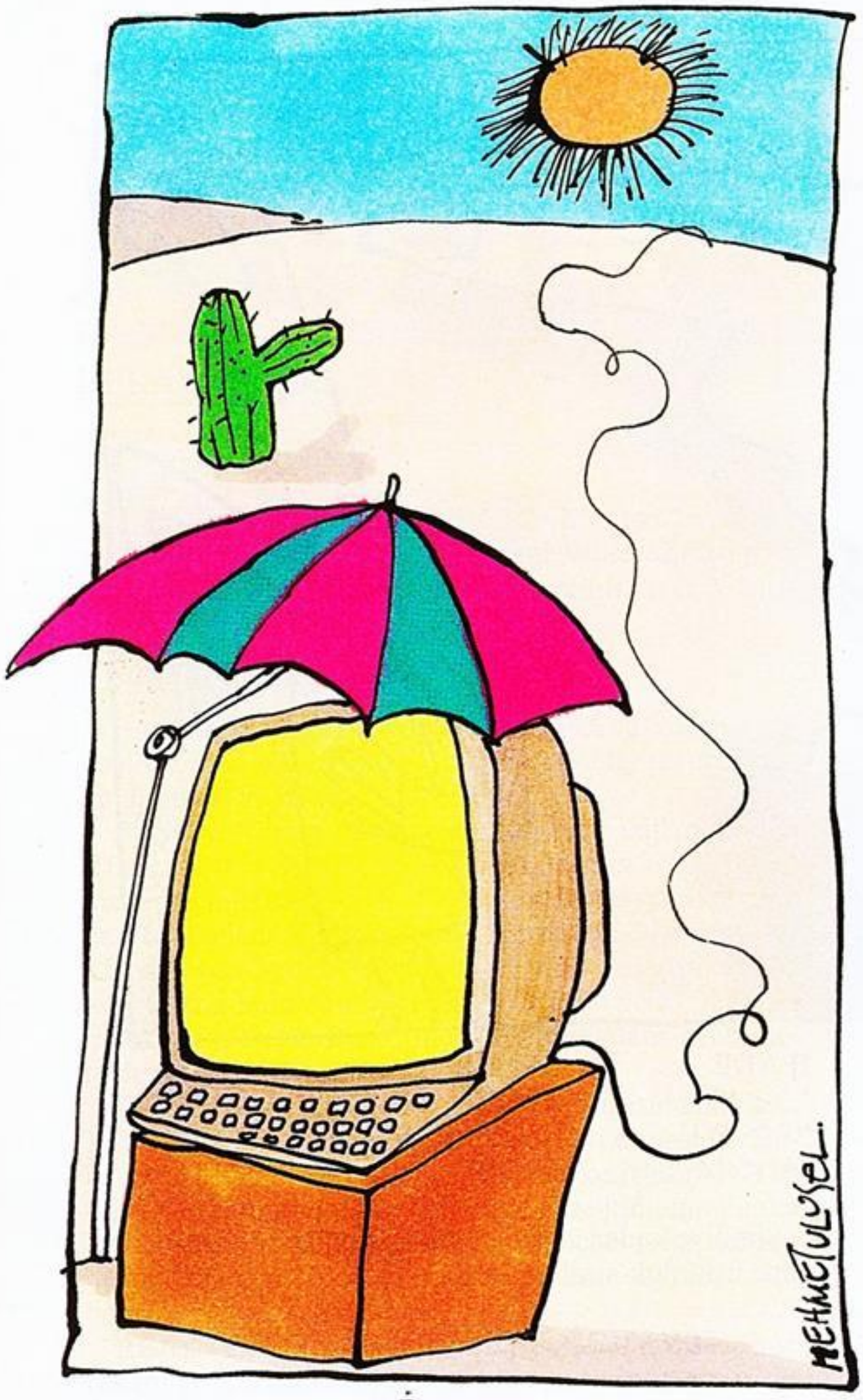
BOOLEAN İŞLEMCİLERİ

Bu işlemciler NOT, OR ve AND'dir. Bu işlemcileri BOOLEAN ifadelerde kullanırız, bu ifadelerin sonucu BOOLEAN olur.

NOT işlemcisinin aşağıdaki kullanılışında elde edilen değerlere dikkat ediniz.

Ifade	Deger
NOT TRUE	FALSE
NOT FALSE	TRUE

Aşağıda OR ve AND işlemcilerinin doğruluk tablosu yeralmaktadır.



Ifade	Deger
TRUE OR TRUE	TRUE
TRUE OR FALSE	TRUE
FALSE OR TRUE	TRUE
FALSE OR FALSE	FALSE
TRUE AND TRUE	TRUE
TRUE AND FALSE	FALSE
FALSE AND TRUE	FALSE
FALSE AND FALSE	FALSE

Yeni öğrendiğimiz işlemcilerle birlikte işlemcilerin üstünlük sıralamasını tekrar oluşturalım.

	İlk yapılacak
NOT	
/ , * , DIV , MOD ,	
AND + , — , OR	
= , < , > , < = ,	
> =, < >	

OR, AND ve NOT işlemcilerinizi komutlarda kullanırken ifade de parantez kullanmanız çok iyi olacaktır, böylece meydana gelebilecek herhangi bir karışıklığı önlemiş olacaksınız.

Örnek: (12 4*5+2) OR (5=2) AND NOT (3+3=6)

Yazımızın bundan sonraki kısmında şimdiye kadar üç bölümde anlattıklarımızı kısaca gözden geçireceğiz. Bundan sonraki bölümden itibaren PASCAL dilinde yeralan tekrarlama komutlarını anlatmaya başlayacağız.

İlk önce aşağıdaki altı maddeyi hiçbir zaman unutmayınız.

1. Bilgisayar hafızası hiçbir zaman boş değildir, fakat hafızanın ilk içeriği sizin programınız için anlamsız olabilir.

2. Hafızadaki bir bilgi o yere yeni bir bilgi konduğunda yok olacaktır (mesela REDLN komutu ile bir veri okutursanız).

3. Bir programı çalıştırmak için o programı önce bilgisayarınızın hafızasına yüklemelisiniz.

4. Herhangi bir verinizin işlenebilmesi için ilk önce o verinin hafızaya yazılması gerekir.

5. Bir bilgisayar kendi başına düşünemez, bilgisayarın bir işi yapabilmesi için ona önce o işi yaptıracak olan programı yüklemeliyiz.

6. Bilgisayarı programlamak çok eğlenceli olacaktır, fakat öncelikle sabırlı, düzenli ve dikkatli olmalısınız.

Programlarınızda kullanacağınız her değişken, sabit belirteçleri ilk önce tanımlamayı unutmayın, unutmamanız gereken bir değer şey de komutları noktalı virgülle ";" ayırmaktır.

Şimdiye kadar öğrenmiş olduğumuz PASCAL komutları ve bunların yaptıkları işler şunlardır.

KOMUT	ETKİSİ
Program komutu PROGRAM adres (INPUT, OUTPUT)	Programımızın ismi ADRES, ,veri dosyaları INPUT ve OUTPUT adını almıştır.
Sabit tanımı CONST kdv=0.1; yıldız='*';	kdv belirtecine sabit olarak 0.1 , yıldız belirtecine sabit olarak * atanmıştır.
Değişken tanımı VAR a, b, c: REAL; ı, j: INTEGER;	Hafızada REAL sayılar için a, b, c değişkenlerine INTEGER sayılar için ı ve j değişkenlerine yer ayrılıyor.
Atama komutu kar := satış — alış;	satış değişkeninin değerinden alış değişkinin değeri çıkarılarak sonuç kâr değişkenine atanıyor.
WRITE komutu WRITE('UCRET=', ucret:7:2);	ücret değişkeninin değeri yedilik alanda noktadan sonra iki rakam olmak üzere yazılacaktır.
READLN komutu READLN (hız, zaman);	hız ve zaman değişkenlerine veri girişi yapılır.

Şimdilik hoşçakalın... □

Teleteknik'ten Paketler

Temel Eğitim Türkçe Programları

I/1

- Ses Bilgisi
- Kelimelerde Yapı
- Kelimelerde Anlam

II1

- Zarflar
- Edat-Bağlaç Ünlem
- Cümle

I2

- İsimler
- Sıfatlar
- Zamirler

III1,2,3

Testler

II2

- Fil Çatısı
- Yazım
- Noktalama

I3

- Filler
- Ek-Fiil
- Bileşik Fiilller
- Fiilni siler

II3

- Deyimler
- Atasözleri
- Nesir
- Nazım

Bilçağ ve Türkçe eğitim paketlerini Temmuz 1987 tarihine kadar alanlar arasında yapılacak çekilişle şu hediyeler verilecektir.

- 1 Adet Renkli Monitör
- 2 Adet Yeşil Monitör
- 5 Adet Plotter
-20 Adet Orijinal Kitap
-20 Adet Orijinal Kaset
-10 Adet Magıc Voice Set.
-20 Adet Programcının el kitabı
-20 Adet Basıc'le Programlama (Kitap)
-10 Adet SYS-CONT joystick.

Commodore

Amiga günü.

Gün, Amigaların günü! Çünkü uzun zamandır yalnızca hayal edilebilen olanakları, nihayet, Amiga sunuyor!
Amiga'nın, duyanları olağanüstü heyecanlandıran eşsiz yeteneklerinden bazıları, huzurlarınızda!

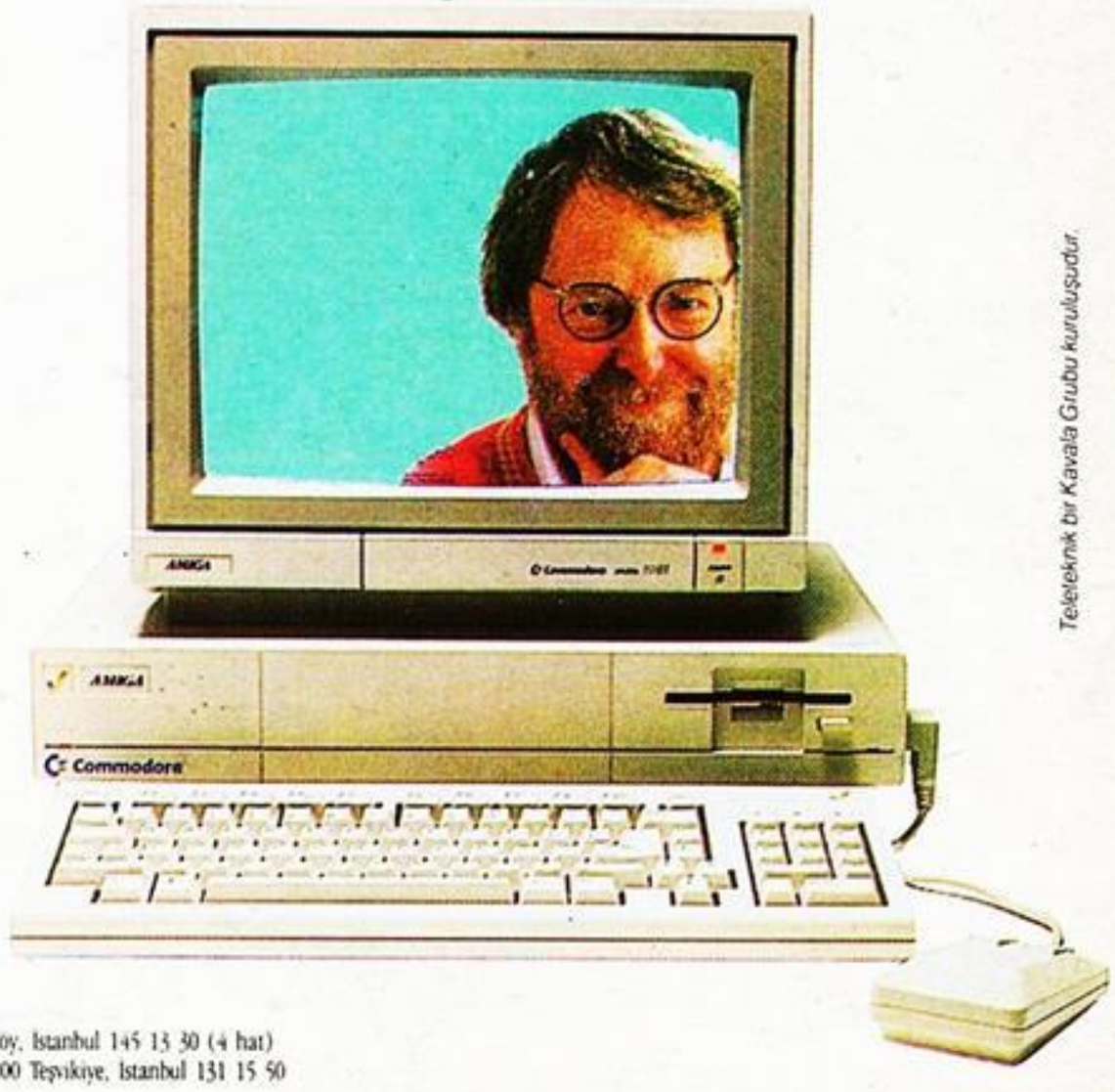
● Durağan görüntüleri hareketli görüntülere Amiga ile dönüştürebiliyorsunuz. ● Banttan ya da kameradan doğrudan görüntü alabiliyor; monitördeki görüntüyü, video kasete kaydedebiliyorsunuz. ● 4 ayrı sesi stereo olarak üretebiliyorsunuz. ● İnsan sesi üretebiliyorsunuz. ● Sayısız müzik aletini Amiga ile çalabiliyor, dilerseniz orkestra bile kurabiliyorsunuz. ● Elde ettiğiniz görüntü ile senkron, müzik yapabiliyorsunuz. ● Amiga'daki renk seçeneği sayısı, tam 4096! ● Amiga'da, görüntüleri renkli fotoğraf kalitesinde saklamak mümkün. ● Muazzam bellek, 32 Bit hız, Amiga'da! ● Multi tasking işletim sistemi sayesinde, aynı anda çok sayıda programda birden çalışabilme olanağını, size, Amiga sunuyor.

Grafikerler! Mimarlar! Mühendisler! Doktorlar! Tasarımcılar! Ve süper bir bilgisayardan yararlanabilecek herkes!
Gün, Amigaların günü! Yıl, Commodore'un yılı! Yararlanın!

AMIGA

Profesyonellerin Aradığı Üstünlükler
Commodore Amiga'da Standard

Merkezi İşlem Birimi	: MC 68000 (8 MHZ, 16/32 Bit)
Yardımcı İşlem Birimleri	: Custom Animation Custom Graphics Custom Sound/Peripherals
Ana Bellek	: 512 KB Standart (8,5 MB'a Kadar Arttırma İmkanı)
Ara Birimler	: - Mouse, Grafik Tablet, Lightpen, Joystick ve Paddle için 2 Bağlantı Yuvası - Bellek arttırma, harddisk bağlantısı için genişleme yuvası - Programlanabilir centronics paralel port - RS232C (V24) seri port - Stereo ses çıkışı - RGB analog ve digital çıkışlar - 3 adet 3,5 veya 5,25 inch disket sürücü bağlantısı için disket sürücü çıkışı
Klavye	: Düşük yükseklik, eğimi ayarlanabilir, bağımsız 10 fonksiyon, HELP, 4 cursor hareket ve 15 numerik tuş olmak üzere toplam 89 tuşa sahiptir.
Ekran	: - Renkli grafik 60 veya 80 kolon - 16 renk 640 x 400 yüksek çözümleme gücü - 4096 renk paleti - 8 adet 16 bit genişliğinde yaratık
Yazılım	: - Amiga DOS Multitasking UNIX benzeri, Mouse veya Keyboard kontrolu - Amiga Basic - Amiga Tutorial - Kaleidoscope - Voice Synthesis Library
Ek Birimler	: - Dotmatrix, LQ, renkli bilgiyazarlar - Commodore PC uyumlu ek donanım Sidecar - 10/20 MB harddisk - Video digitizer-Frame grabber, Genlock - MIDI interface



Teleteknik bir Kavala Grubu kuruluşudur.

Teleteknik

Merkez: Rıhtım Cad. Neşli Han 207/4, 80030 Karaköy, İstanbul 145 13 30 (4 hat)
Bilgisayar Danışma Merkezi: Silahhane Cad. 59 80200 Teşvikiye, İstanbul 131 15 50

COMMODORE
AYLIK BİLGİSAYAR DERGİSİ Mayıs 1987

Okur Anketi

Sevgili Commodore Dostu,
Başta dergimiz olmak üzere, TELETEKNİK'in vermekte olduğu tüm hizmetlerin daha verimli olmasını sağlamak amacıyla ''Commodore Dostları''nın doldurup göndermesini beklediğimiz bir küçük anket yayınlıyoruz. Anket formlarını bize en kısa zamanda ulaştıracağınıza inancımız tamdır. Anket sorularını tam ve sağlıklı biçimde doldurmanızı, özellikle dergi üzerinize düşünce ve beklentilerinizi ayrıntılı bir biçimde belirterek, ''Commodore Dergisi, Rıhtım Cad. 207 Nesli Han, K.4 Karaköy/İst.'' adresine göndermenizi rica ediyoruz.
Katılımınız için şimdiden teşekkür eder, Commodore dolu günler dileriz...
Posta ücreti tarafımızdan ödenecektir. Lütfen pul yapıştırmayınız.

1. Bilgisayarınızın marka ve modeli nedir?

Commodore PC-10/PC-20	☐
Commodore 8032/8096/8296/8296D	☐
Commodore 128D	☐
Commodore 128	☐
Commodore 64	☐
Commodore SX-64	☐
Commodore Plus-4	☐
Commodore 16	☐
Diğer Marka/Model (Yazınız)	.....
Bilgisayarınız yok	☐ -Soru 4'e geçiniz.

2- a) Bilgisayar dışında kullandığınız diğer donanım malzemeleri nelerdir?
b) Almayı düşündüğünüz başka donanım malzemeleri var mı? Varsa nelerdir?

	a	b
Teyp	☐	☐
Disket Sürücü	☐	☐
Printer	☐	☐
Plotter	☐	☐
Yeşil Monitör	☐	☐
Renkli Monitör	☐	☐
S/B TV	☐	☐
Renkli TV	☐	☐
Diğer (Yazınız)	.....	.....
Yok	☐	☐

3- a) Bilgisayarınızı alırken ne amaçla kullanmayı düşünüyordunuz?
b) Şu anda ne amaçla kullanıyorsunuz?

	a	b
Temel Eğitim Amacıyla	☐	☐
Oyun Amacıyla	☐	☐
İş Amacıyla	☐	☐
Yabancı Dil Öğrenimi	☐	☐
Program Yazma	☐	☐
Bilgisayar Dili Öğrenimi (Basıc, Cobol, Pascal)	☐	☐
Diğer (Yazınız)	.....	.....

4- Commodore kullanıcısı olarak en büyük sorununuz nedir?

.....
.....
.....

5- Lütfen Commodore'da yer alan yazıların değerlendirmesini yapınız!

	Çok İyi	Yeterli	Orta	Yetersiz
ÖZEL BÖLÜM	☐	☐	☐	☐
DONANIM	☐	☐	☐	☐
Alet Edevat	☐	☐	☐	☐
Diğer Yazılar	☐	☐	☐	☐
YAZILIM	☐	☐	☐	☐
Kaset-Disket	☐	☐	☐	☐
Diğer Yazılar	☐	☐	☐	☐
TANITIM	☐	☐	☐	☐
Oyunlar	☐	☐	☐	☐
Eğitim Programları	☐	☐	☐	☐
Compi-Hobi Compi-Fobi	☐	☐	☐	☐
Okur Mektupları	☐	☐	☐	☐
Yazışma	☐	☐	☐	☐
Çizgi Köşesi	☐	☐	☐	☐
Püf Noktası	☐	☐	☐	☐
Küçük İlanlar	☐	☐	☐	☐
Diğer Yazılar	☐	☐	☐	☐
PROGRAM DÖKÜM EKİ	☐	☐	☐	☐

6- Derginizdeki reklamlardan yararlanıyor musunuz?

Evet ☐
Hayır ☐

7- Derginizin düzeyini değerlendirir misiniz?

Düzeyi Yüksek ☐
Orta, yeterli ☐
Düzeyi düşük ☐

8- Genelde derginiz ile ilgili düşünceleriniz, beklentileriniz.

.....
.....
.....

İstatistik

CİNSİYET

Kadın ☐
Erkek ☐

YAŞ

0-10 ☐
10-14 ☐
15-19 ☐
20-24 ☐
25-34 ☐
35-49 ☐
50+ ☐

MESLEK

	Kendinizin	Aile Reisinin
Öğrenci	☐	☐
Ortaokul	☐	☐
Lise	☐	☐
Üniversite	☐	☐
Ev kadını	☐	☐
Serbest Meslek (doktor, avukat vb.)	☐	☐
Sanayici	☐	☐
Memur (Kamu)	☐	☐
Yönetici	☐	☐
Emekli	☐	☐
Diğer (Belirtiniz)	.....	.....

ÖĞRENİM DURUMU (SON BİTİRİLEN EĞİTİM KURUMU)

	Kendinizin	Aile Reisinin
İlkokul	☐	☐
Ortaokul	☐	☐
Lise	☐	☐
Yüksek Okul	☐	☐
Yüksek Lisans/Doktora	☐	☐

MEDENİ DURUMU

Evli ☐
Bekar ☐

HANE HALKI SAYISI

1 Kişi ☐
2 Kişi ☐
3 Kişi ☐
4 Kişi ☐
4+ Kişi ☐

HANEDEKİ ÇOCUKLARIN YAŞLARI

	5-10	10-14	15-19	20-24
1. Çocuk	☐	☐	☐	☐
2. Çocuk	☐	☐	☐	☐
3. Çocuk	☐	☐	☐	☐
4. Çocuk	☐	☐	☐	☐

SAHİP OLDUĞUNUZ EŞYALAR

Otomobil ☐
Buzdolabı ☐
Çamaşır Makinası-Merdaneli ☐
Otomatik ☐
Siyah Beyaz TV ☐
Renkli TV ☐
Video ☐
Müzik Seti ☐
Bulaşık Makinası ☐

HANE HALKI GELİRİ (TOPLAM NET)

-100.000 TL ☐
101.000-150.000 TL ☐
151.000-200.000 TL ☐
201.000-250.000 TL ☐
251.000-300.000 TL ☐
301.000-400.000 TL ☐
500.000+ TL ☐

SOSYO-EKONOMİK GRUP

A ☐
B ☐
(Boş Bırakınız) C ☐

AD-SOYAD: .....

ADRES: .....

.....

.....

.....

Gönderen:

Posta ücreti müessesemizce ödenecektir.

COMMODORE DERGİSİ OKUR ANKETİ

Rıhtım Caddesi No: 207/4 Nesli Han 80030 Karaköy-İSTANBUL